KSBi-BIML 2024

Bioinformatics & Machine Learning(BIML)
Workshop for Life and Medical Scientists

생명정보학 & 머신러닝 워크샵 (온라인)

Single-cell RNA-sequencing analysis: Assignment of cell types

김규태 _ 아주대학교

KSBI KOREAN SOCIETY FOR BIOINFORMATICS | 한국생명정보학회

본 강의 자료는 한국생명정보학회가 주관하는 BIML 2024 워크샵 온라인 수업을 목적으로 제작된 것으로 해당 목적 이외의 다른 용도로 사용할 수 없음을 분명하게 알립니다.

이를 다른 사람과 공유하거나 복제, 배포, 전송할 수 없으며 만약 이러한 사항을 위반할 경우 발생하는 **모든 법적 책임은 전적으로 불법 행위자 본인에게 있음을 경고**합니다.

KSBi-BIML 2024

Bioinformatics & Machine Learning(BIML)
Workshop for Life and Medical Scientists

안녕하십니까?

한국생명정보학회가 개최하는 동계 교육 워크샵인 BIML-2024에 여러분을 초대합니다. 생명정보학 분야의 연구자들에게 최신 동향의 데이터 분석기술을 이론과 실습을 겸비해 전달하고자 도입한 전문 교육 프로그램인 BIML 워크샵은 2015년에 시작하여 올해로 벌써 10년 차를 맞이하게 되었습니다. BIML 워크샵은 국내 생명정보학 분야의 최초이자 최고 수준의 교육프로그램으로 크게 인공지능과 생명정보분석 두 개의 분야로 구성되어 있습니다. 올해 인공지능 분야에서는 최근 생명정보 분석에서도 응용이 확대되고 있는 다양한 인공지능 기반 자료모델링 기법들에 대한 현장 강의가 진행될 예정이며, 관련하여 심층학습을 이용한 단백질구조예측, 유전체분석, 신약개발에 대한 이론과 실습 강의가 함께 제공될 예정입니다. 또한 단일세포오믹스, 공간오믹스, 메타오믹스, 그리고 롱리드염기서열 자료 분석에 대한 현장 강의는 많은 연구자의 연구 수월성 확보에 큰 도움을 줄 것으로 기대하고 있습니다.

올해 BIML의 가장 큰 변화는 최근 연구 수요가 급증하고 있는 의료정보자료 분석에 대한 현장 강의를 추가하였다는 것입니다. 특히 의료정보자료 분석을 많이 수행하시는 의과학자 및 의료정보 연구자들께서 본 강좌를 통해 많은 도움을 받으실 수 있기를 기대하고 있습니다. 또한 다양한 생명정보학 분야에 대한 온라인 강좌 프로그램도 점차 증가하고 있는 생명정보 분석기술의 다양화에 발맞추기 위해 작년과 비교해 5강좌 이상을 신규로 추가했습니다. 올해는 무료 강좌 5개를 포함하여 35개 이상의 온라인 강좌가 개설되어 제공되며, 연구 주제에 따른 연관된 강좌 추천 및 강연료 할인 프로그램도 제공되며, 온라인을 통한 Q&A 세션도 마련될 예정입니다. BIML-2024는 국내 주요 연구중심 대학의 전임 교원이자 각 분야 최고 전문가들의 강의로 구성되었기에 해당 분야의 기초부터 최신 연구 동향까지 포함하는 수준 높은 내용의 강의가 될 것이라 확신합니다.

BIML-2024을 준비하기까지 너무나 많은 수고를 해주신 운영위원회의 정성원, 우현구, 백대현, 김태민, 김준일, 김상우, 장혜식, 박종은 교수님과 KOBIC 이병욱 박사님께 커다란 감사를 드립니다. 마지막으로 부족한 시간에도 불구하고 강의 부탁을 흔쾌히 허락하시고 훌륭한 현장 강의와 온라인 강의를 준비하시는데 노고를 아끼지 않으신 모든 강사분들께 깊은 감사를 드립니다.

2024년 2월

한국생명정보학회장 이 인 석

강의개요

Single-cell RNA-sequencing analysis: Assignment of cell types

본 강의는 단일세포 전사체 데이터 분석의 기본적인 측면을 다룬다. 단일세포 수준으로 분석하는 것이 왜 중요한지에 대한 개론을 제공하며, 데이터 유형의 구조와 형식을 설명하고, 데이터 전처리 과정을 이해할 수 있도록 이론과 함께 실습 강의를 제공한다. 또한, 단일세포 전사체 데이터를 이용한 세포 유형을 결정하는 전반적인 과정을 이해할 수 있다. 이를 통해 학습자들은 단일세포 연구에서 데이터를 처리하고 세포 유형을 파악하는데 필요한 기초적인 지식을 습득하게 된다.

강의는 다음의 내용을 포함한다:

- 단일세포 전사체 데이터 분석의 중요성과 의의를 이해
- 단일세포 전사체 데이터의 구조와 형식에 대해 학습
- 단일세포 전사체 데이터를 활용하여 세포 유형을 할당하는 과정을 이해

*참고강의교재:

.

* 교육생준비물:

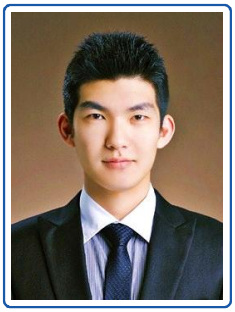
Rstudio 및 Seurat (R package)가 설치된 노트북 (메모리 8GB 이상, 디스크 여유공간 30GB 이상)

* 강의 난이도: 초급

* 강의: 김규태 교수 (아주대학교의과대학 생리학교실) / OOO 조교

Curriculum Vitae

Speaker Name: Kyu-Tae Kim, Ph.D.



▶ **Personal Info**

Name	Kyu-Tae Kim
Title	Assistant Professor
Affiliation	Ajou University School of Medicine

▶ **Contact Information**

Address	164, Wolrd cup-ro, Yeongtong-gu, Suwon 16499
Email	kimqtae@ajou.ac.kr
Phone Number	031-219-4505

Research Interest

Immunogenomics, Cancer evolution, Computational Biology

Educational Experience

2010	B.S., Konkuk University, Seoul, Korea
2012	M.S., Seoul National University, Seoul, Korea
2015	Ph.D., Seoul National University, Seoul, Korea

Professional Experience

2013-2017	Researcher, Samsung Genome Institute, Samsung Medical Center, Seoul, Korea
2017-2019	Postdoctoral Fellow, New York Genome Center, NY, USA
2020-	Assistant Professor, Ajou University School of Medicine, Suwon, Korea

Selected Publications (5 maximum)

1. Determinants of Response and Intrinsic Resistance to PD-1 Blockade in Microsatellite Instability-High Gastric Cancer, Cancer Discovery, 2021 (corresponding author)
2. Somatic mutations and cell identity linked by Genotyping of Transcriptomes, Nature, 2019 (first author)
3. SIDR: simultaneous isolation and parallel sequencing of genomic DNA and total RNA from single cells, Genome Research, 2018 (first author)
4. Application of single-cell RNA sequencing in optimizing a combinatorial therapeutic strategy in metastatic renal cell carcinoma, Genome Biology, 2016 (first author)
5. Single-cell mRNA sequencing identifies subclonal heterogeneity in anti-cancer drug responses of lung adenocarcinoma cells, Genome Biology, 2015 (first author)

KSBI | 한국생명정보학회

KSBi-BIML 2024

Single-cell RNA-sequencing analysis: Assignment of cell types (part1)

Kyu-Tae Kim
Ajou University School of Medicine

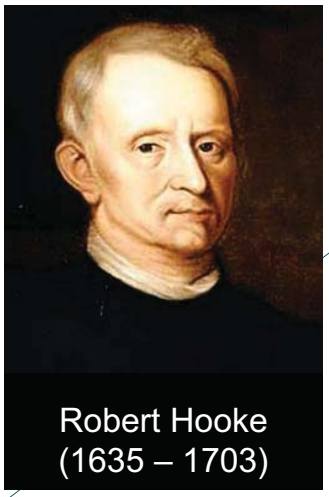
본 교육의 목표와 특징

단일세포 전사체 데이터 전분석

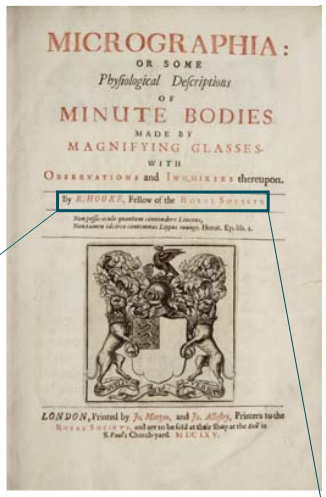
- 단일세포 전사체 데이터 분석의 의의를 이해한다.
- 단일세포 전사체 데이터의 구조와 형식을 이해한다.
- 단일세포 전사체 데이터의 전분석 과정을 이해한다.
- 단일세포 전사체 데이터 normalization 과정을 이해한다.
- 단일세포 전사체 데이터 batch 제거 과정을 이해한다.

Cell: The basic unit of life

Robert Hooke was the first
to apply the word 'Cell' to biological objects (Cork).

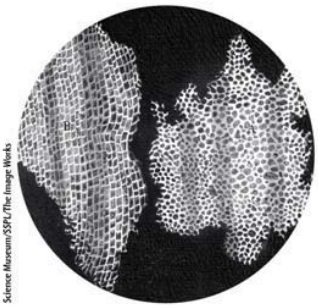


Robert Hooke
(1635 – 1703)



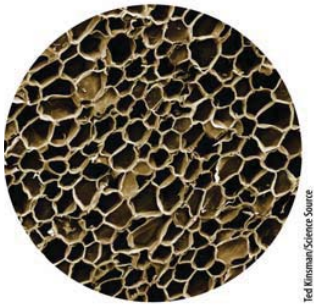
By *R. HOOKE*, Fellow of the ROYAL SOCIETY.

Drawing by Hooke



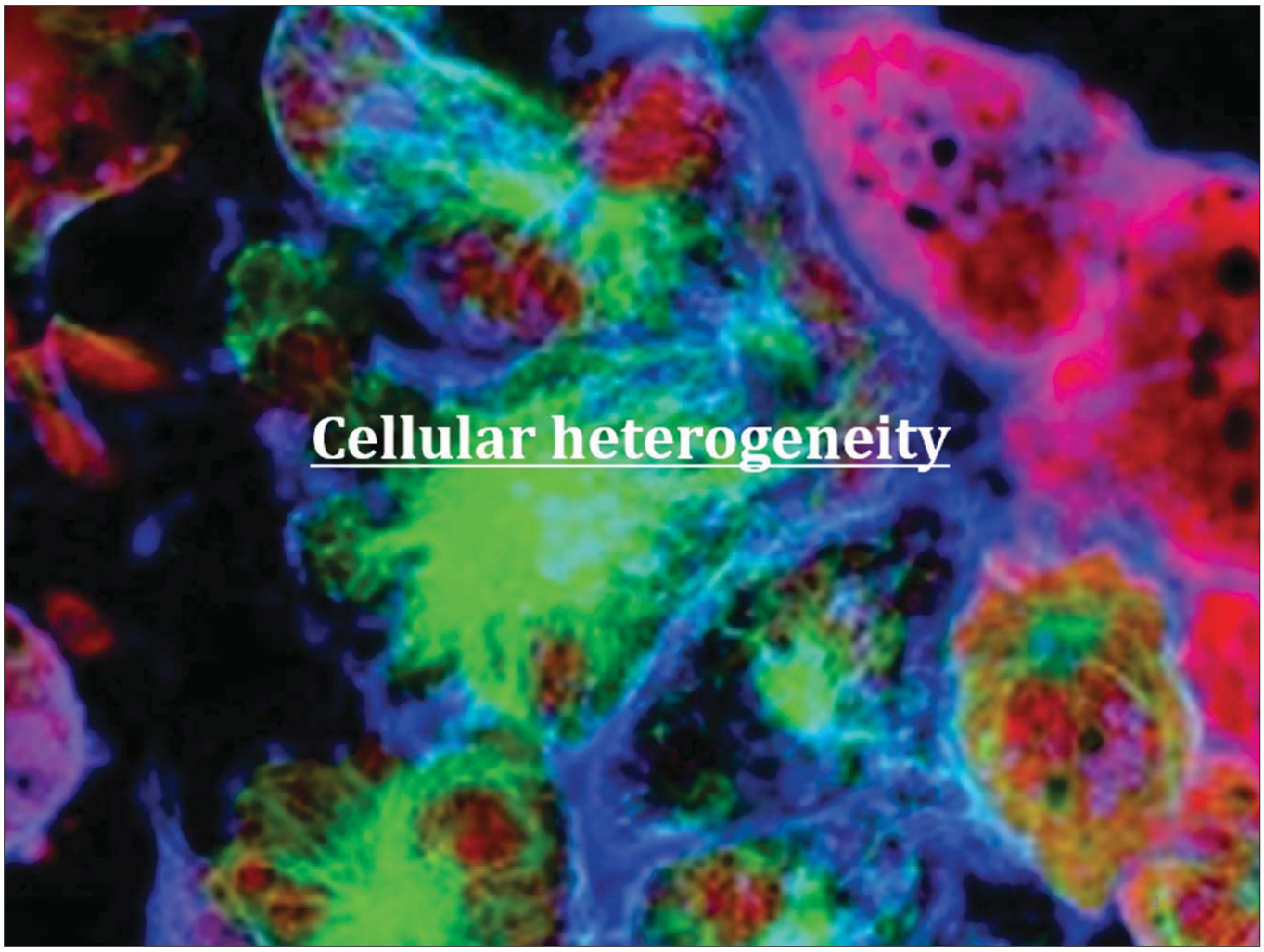
Science Museum/SSPL/The Image Works

Captured picture of Cork tissue



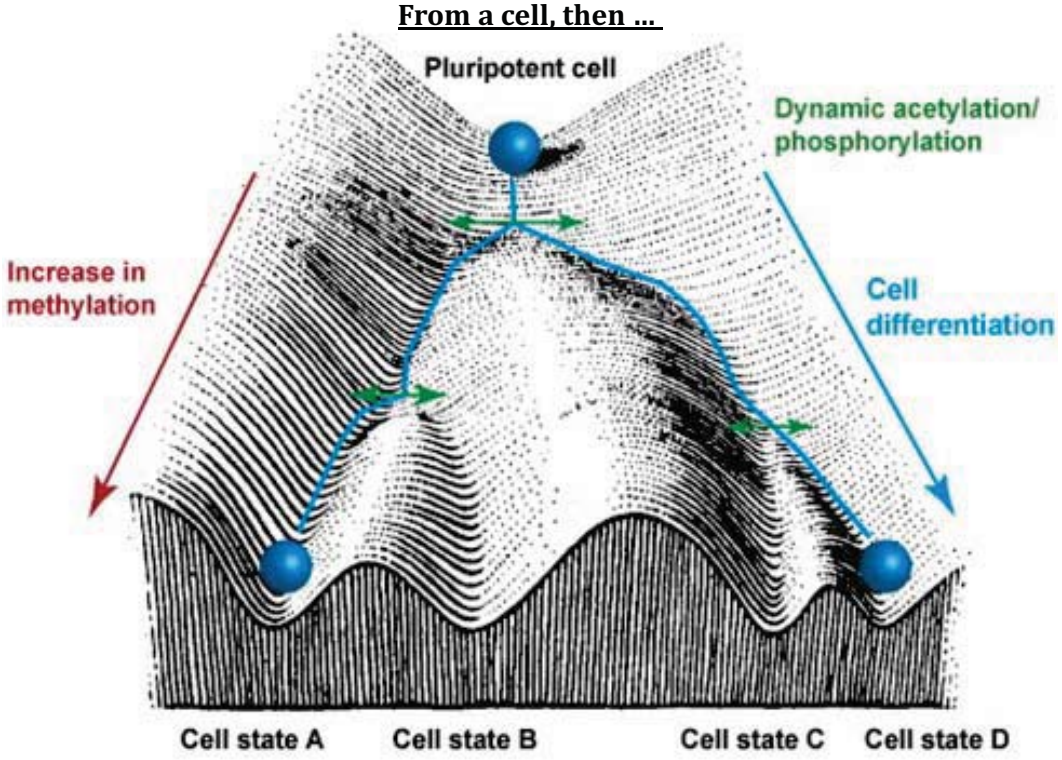
Ted Kinsman/Science Source

3



Cell: The basic unit of life

From a cell, then …



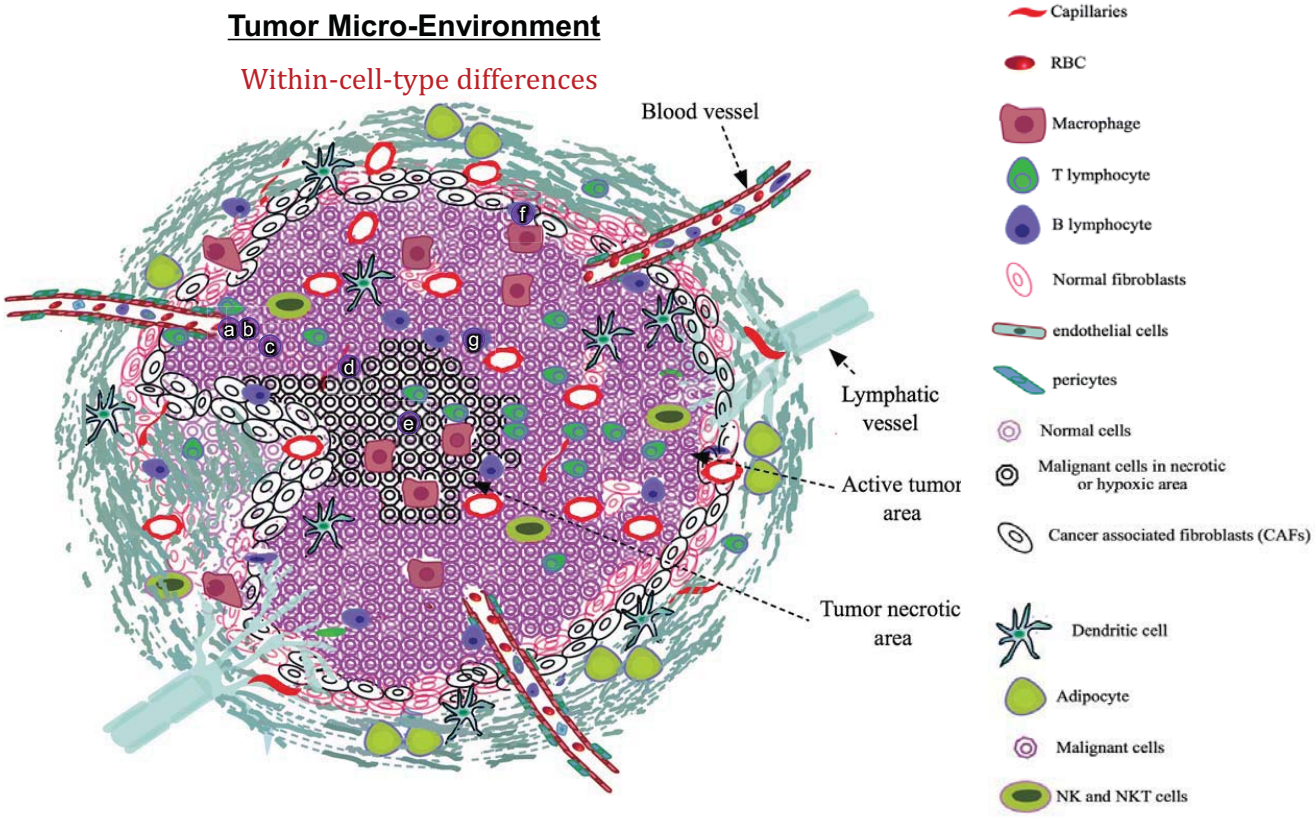
Waddington's model

5

Cell: The basic unit of life

Tumor Micro-Environment

Within-cell-type differences



6

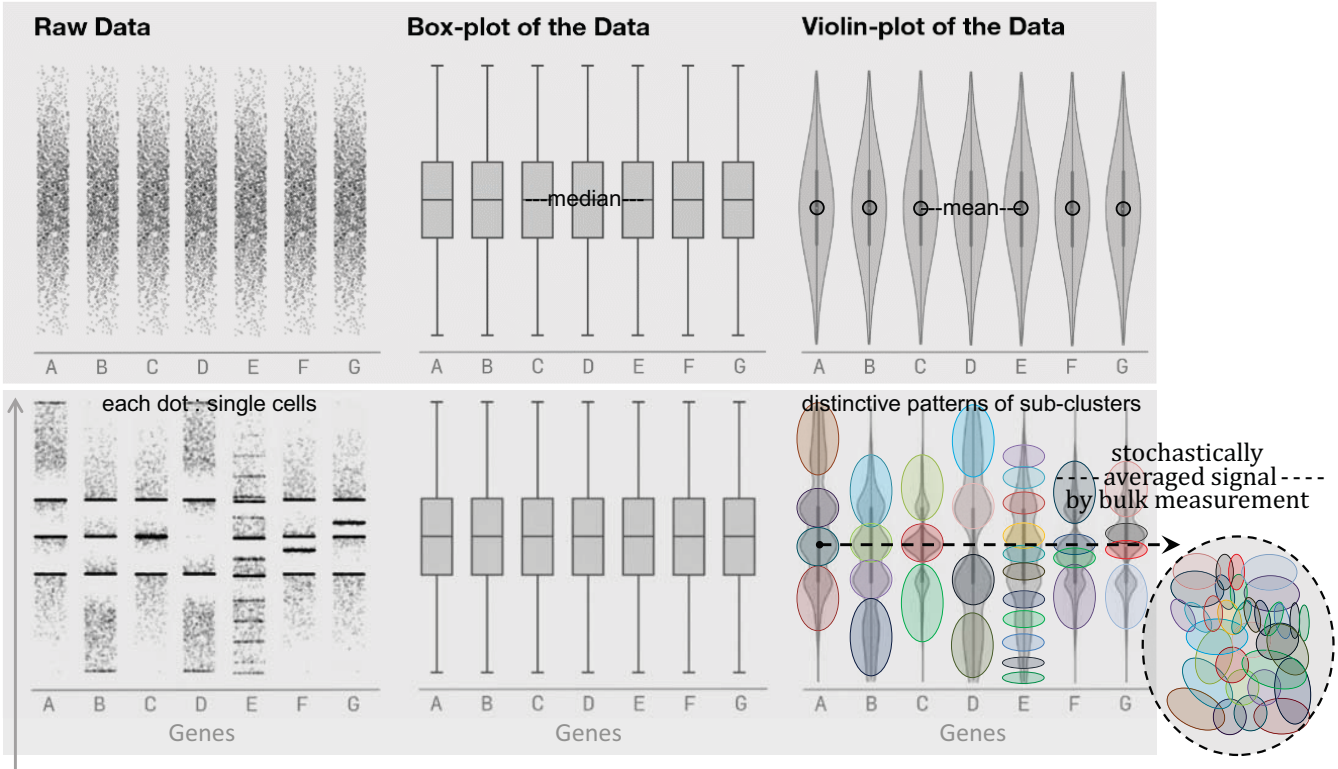
Why single-cell sequencing?

Bulk analysis vs. Single-cell RNA-seq

7

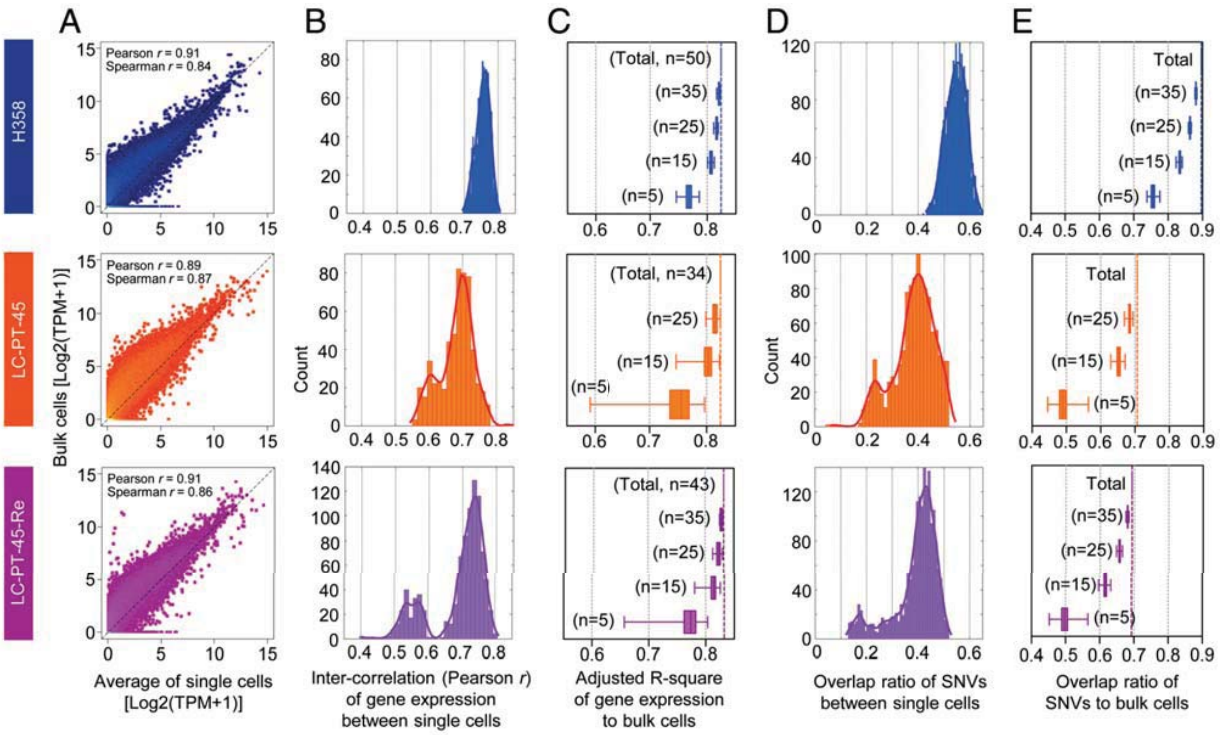
"No! Sometimes the Sum of the Parts (single-cells) is Greater than the Whole (bulk)."

(original phrase by Aristotle, "The Whole is Greater than the Sum of its Parts.")



8

The bulk measurement is
the stochastic average of single cells

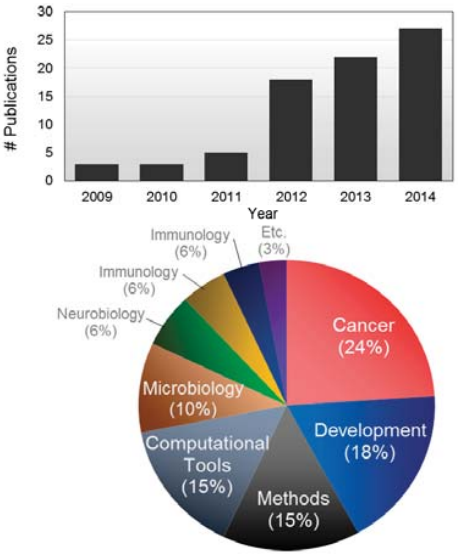


Kim KT, Lee HW, Lee HO et al., 2015 *Genome Biol.*

Single-cell analysis – a brief history

'Single-cell sequencing'
Methods of the Year 2013

Rapid progress in
'Single-cell sequencing'

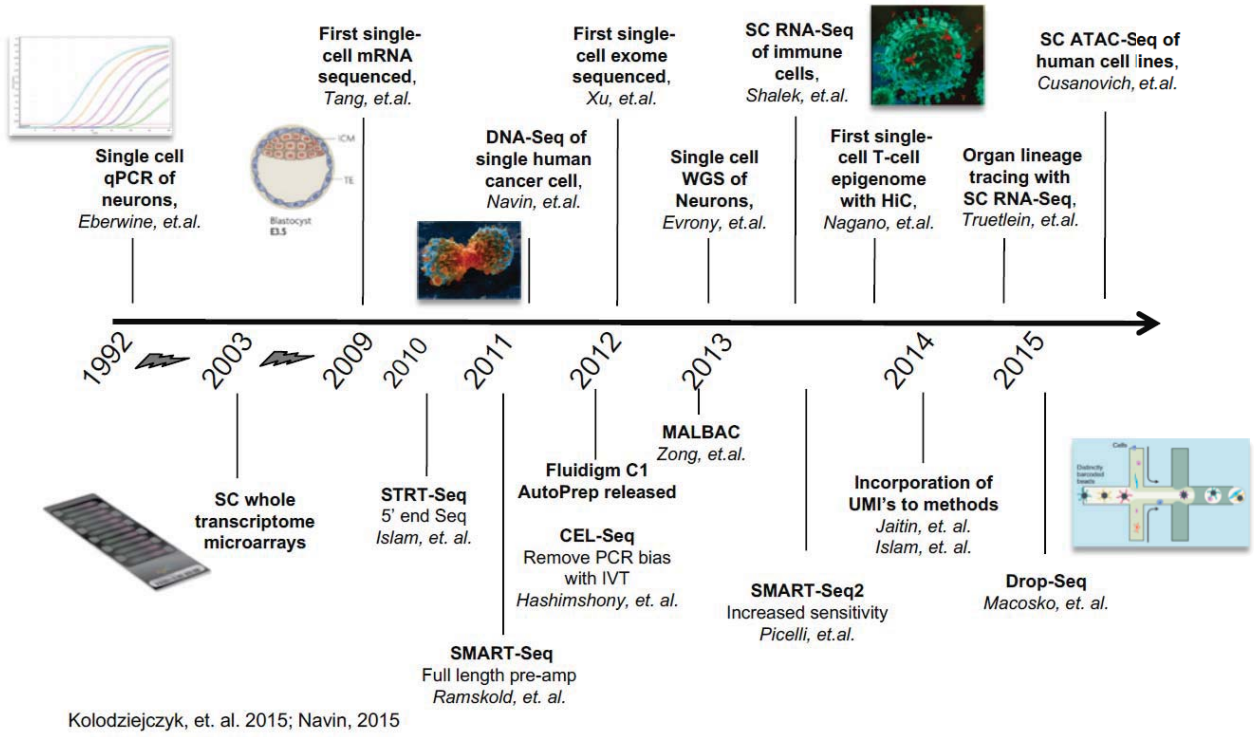


adapted from Wang *et al. Mol Cell* 2015

[Tracking development cell by cell]
Breakthrough of the Year
2018 Science

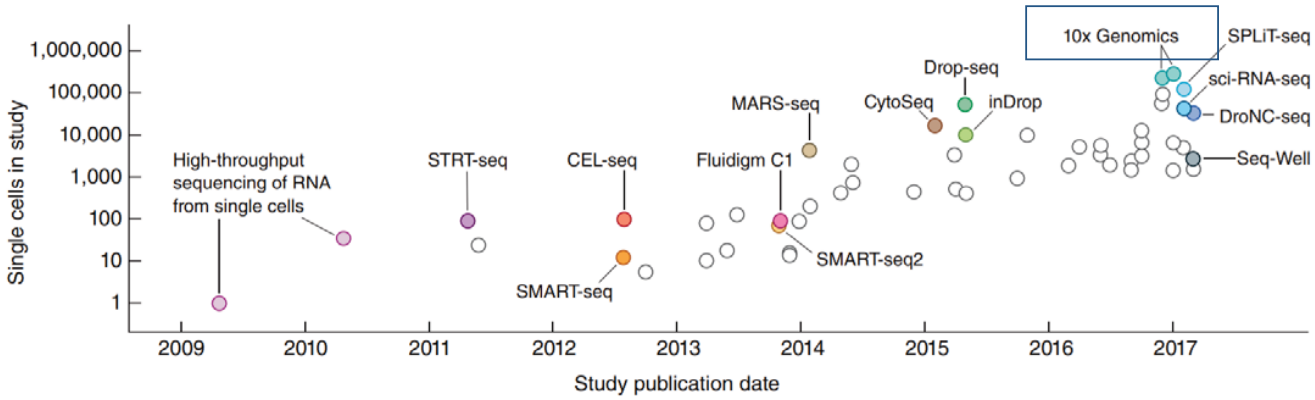
10

Single-cell analysis – a brief history



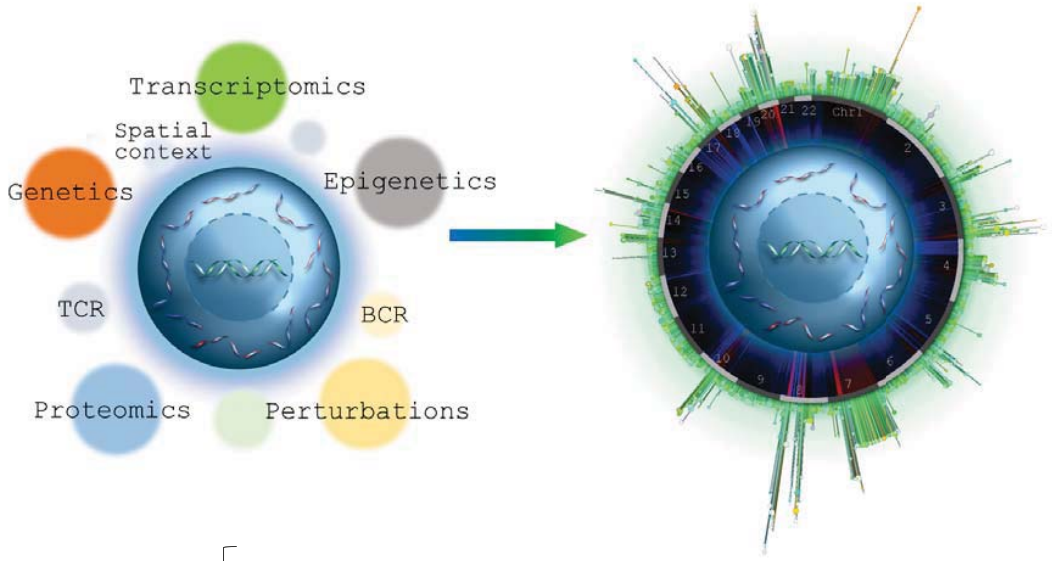
Kolodziejczyk, et. al. 2015; Navin, 2015

Trends: Increasing Dimensionality & More Cells



Sarah Teichmann group, 2018, *Nat Proc.*

[Experimental Approaches]

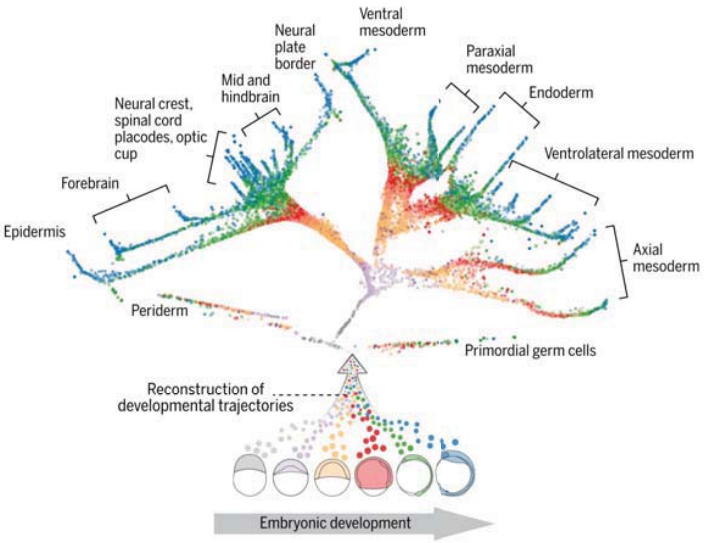


Multi-modal profiling methods at single-cell resolution

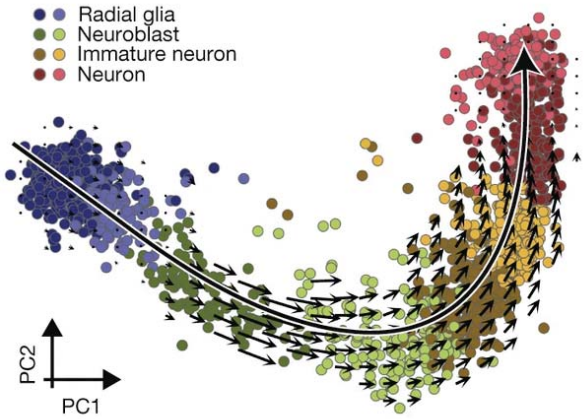
- DNA-seq + RNA-seq = **SIDR-seq**, **G&T-seq**, **DR-seq**
- DNA-seq + RNA-seq + Methyl-seq = **Trio-seq**
- RNA-seq + ATAC-seq = **sciCAR**
- RNA-seq + TCR/BCR = **(10X) 5'GEX with Immune Cell profiling**
- Epitope-profiling + RNA-seq + = **CITE-seq**
- Genotyping + RNA-seq = **GoT**
- Genetic screening with CRISPR + RNA-seq = **Perturb-seq**
- and……

13

[Computational Approaches]



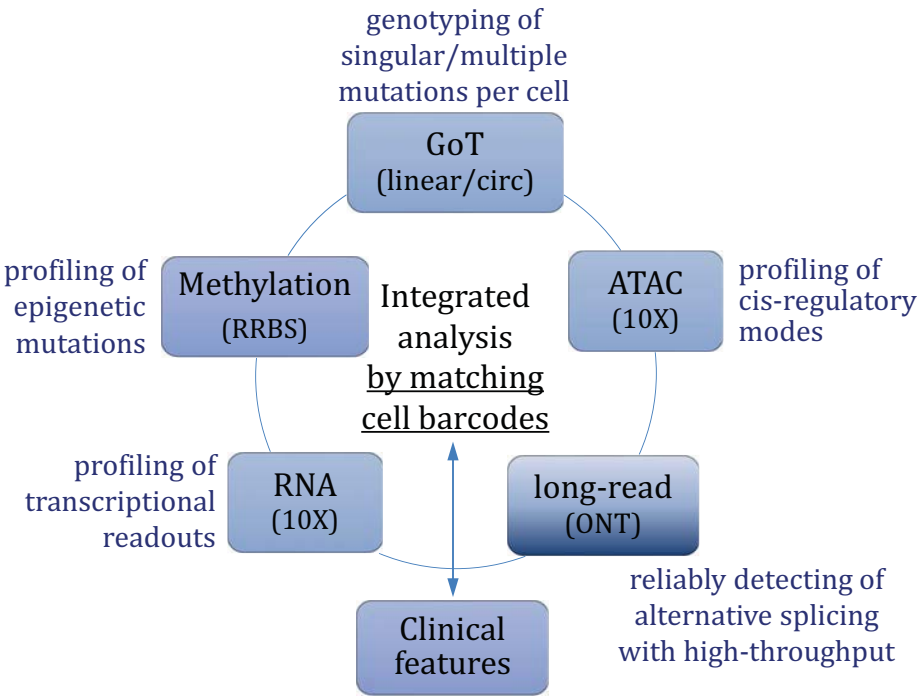
Farrell JA, Wang Y et al., 2018 *Science*



Manno GL et al., 2018 *Nature*

14

[Experimental & Computational Approaches]



15

Highly Dimensional Single-cell Data Sets

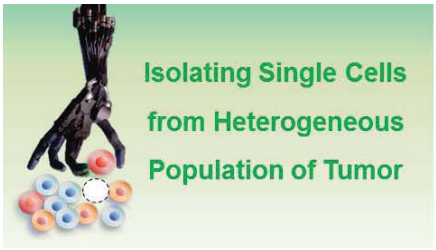
 # Cells x # Features x # Time Points x # Technologies

dissected by

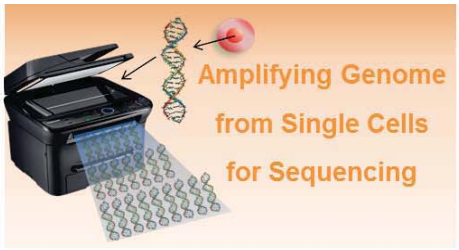
Sophisticated Analytical Design with Massive Computational Power

16

Basic single-cell analysis workflow



- Micropipetting
- Laser capture microdissection
- FACS
- Microfluidic circuits
- Droplet-based microfluidics



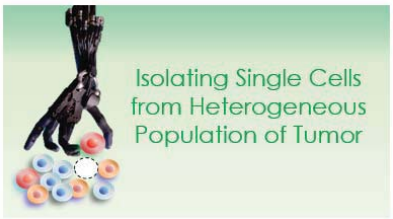
(DNA)
- MALBAC
- MDA
- LIANTI

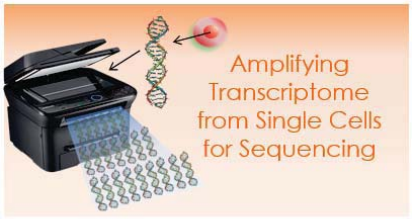
(RNA)
- STRT-seq
- CEL-seq
- SMART/SMART2/SMART3-seq
- Droplet-based amplification (Drop-seq, inDrop, 10X)

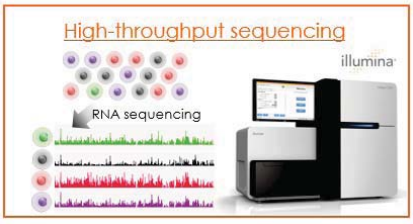
(statistical/algorithmical mining)

17

At the initial stage of single-cell field







96 wells

↓ Dead or live?
↓ Singlet or doublet?

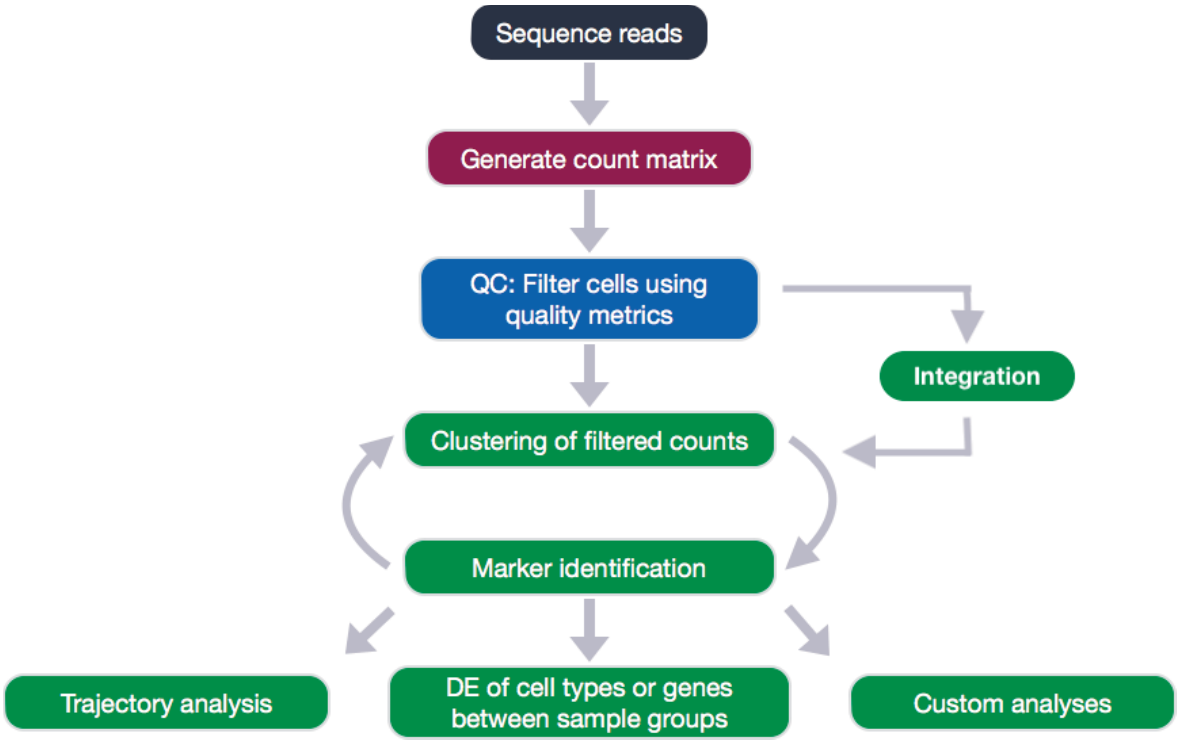
40-80 live cells captured

↓ How many genes detected?
↓ How many reads aligned?
↓ Mitochondria fraction?

20-60 cells of QC-passed

18

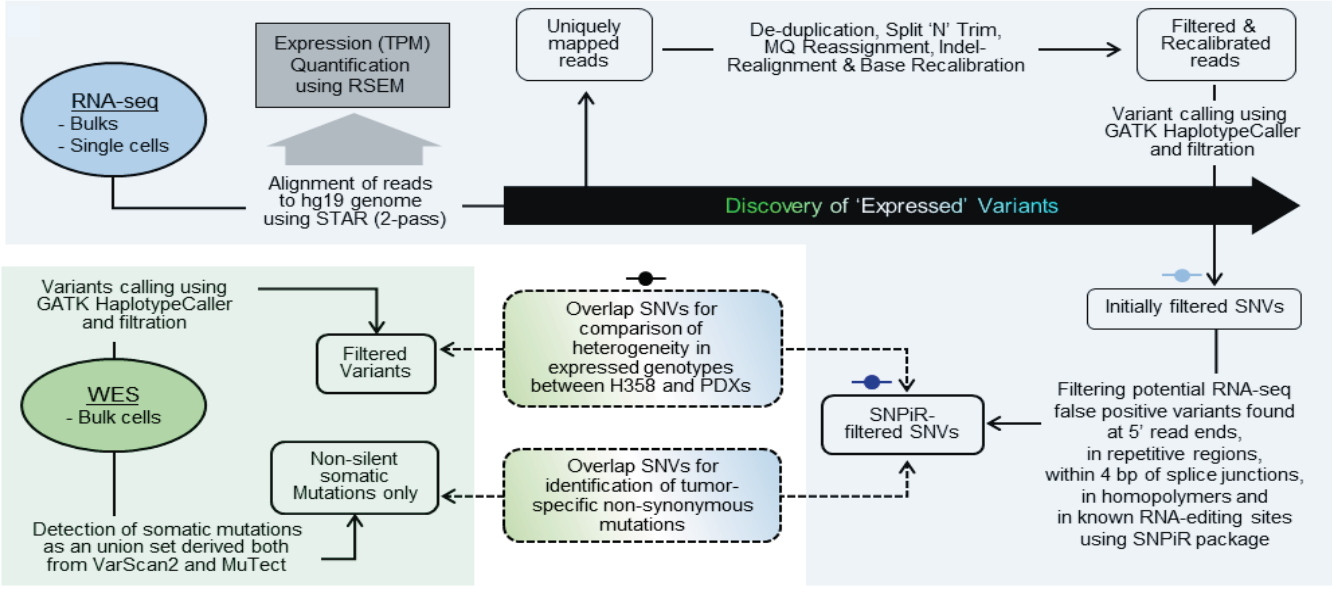
Basic data processing workflow



19

Basic data processing workflow

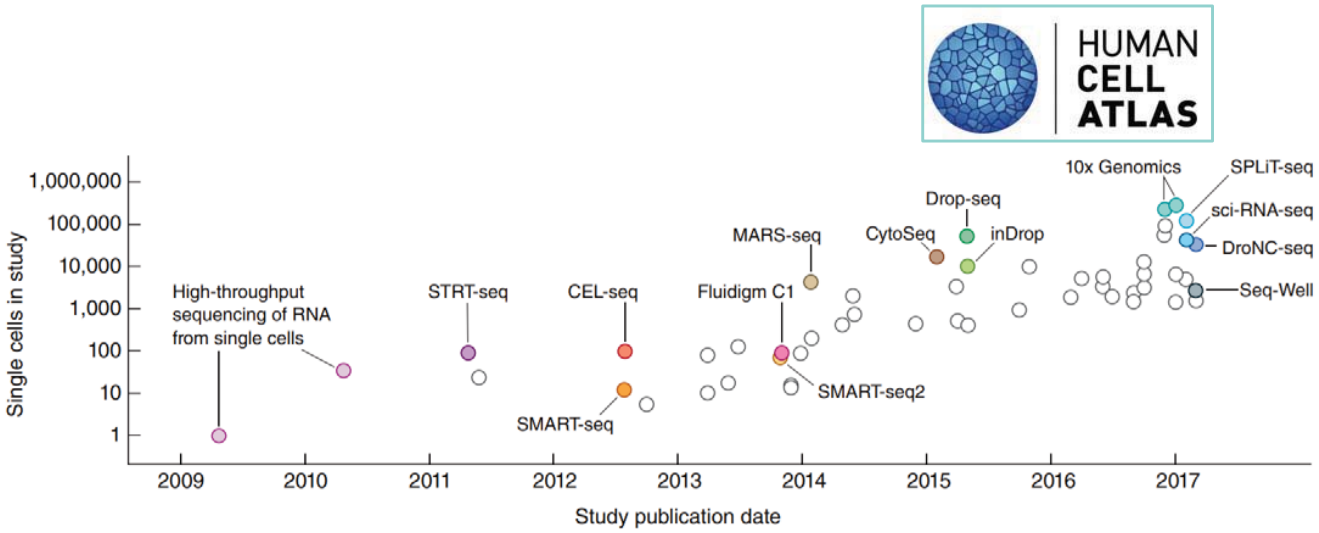
for full-length/high-depth of several single-cells



Kim KT, Lee HW, Lee HO et al., 2015 *Genome Biol.*

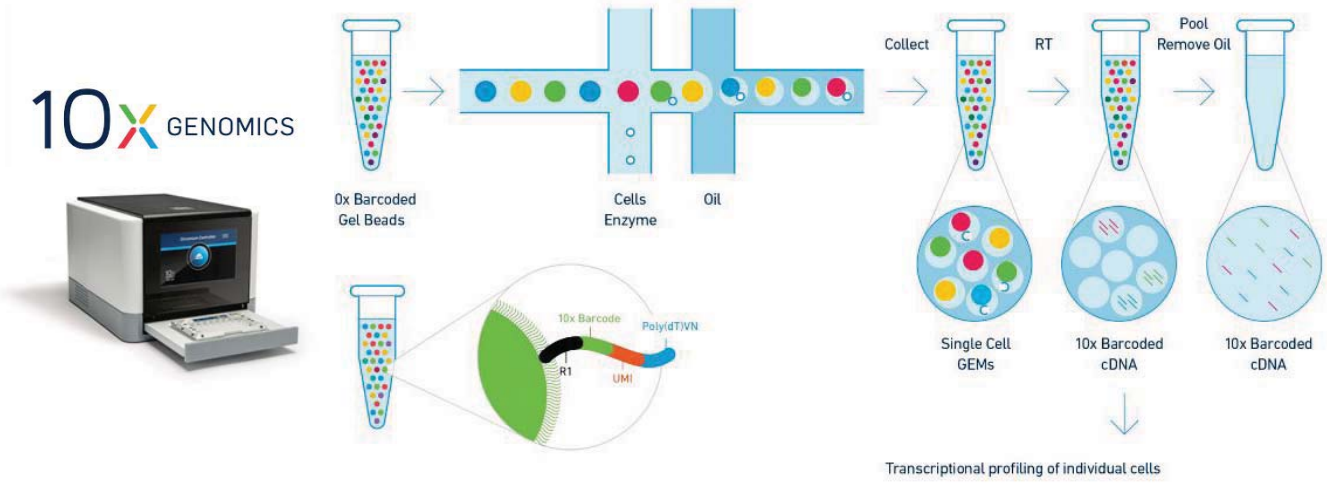
20

Trends: Increasing Dimensionality & More Cells

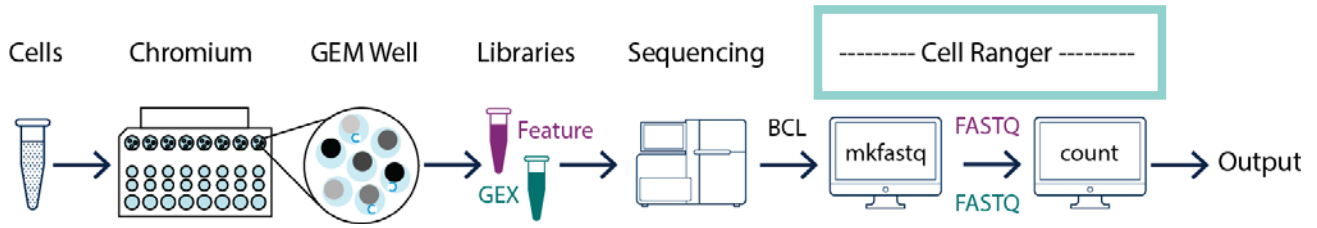


Sarah Teichmann group, 2018, *Nat Proc.*

Basic single-cell analysis workflow



Pre-processing pipeline: 10X CellRanger




```
# /data/users/kimqt2/Projects/chonh_covid19/run_CellRanger.sh
/data/users/kimqt2/program/cellranger-3.1.0/cellranger count \
--id=20_00028_LI_SING \
--fastqs=/data/users/kimqt2/Projects/chonh_covid19/Lung_Fastq/ \
--transcriptome=/data/users/kimqt2/ref/tenX/refdata-cellranger-GRCh38-3.0.0_withSARS_COV2_SNU01 \
--expect-cells=5000 \
--localcores=30 \
--localmem=32
```

--> Output: Gene-level expression matrix per cell

23

CellRanger (10X Genomics)

1. Read Trimming

> Detection/trimming of technically-induced sequence (TSO, template switch oligo)

2. Read Alignment

> Splicing-aware alignment of cDNA sequences to the genome reference using STAR

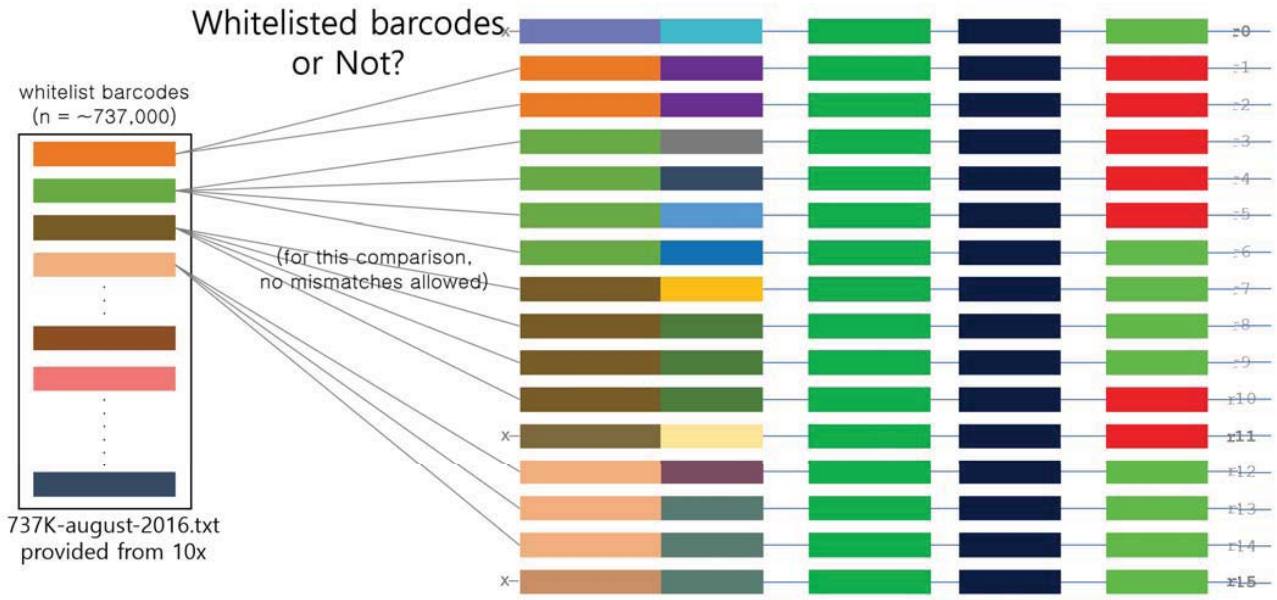
3. Calling cell barcodes and UMI

> Error-aware statistical correction of barcodes and UMI

4. Basic subclustering and dimensional reduction

24

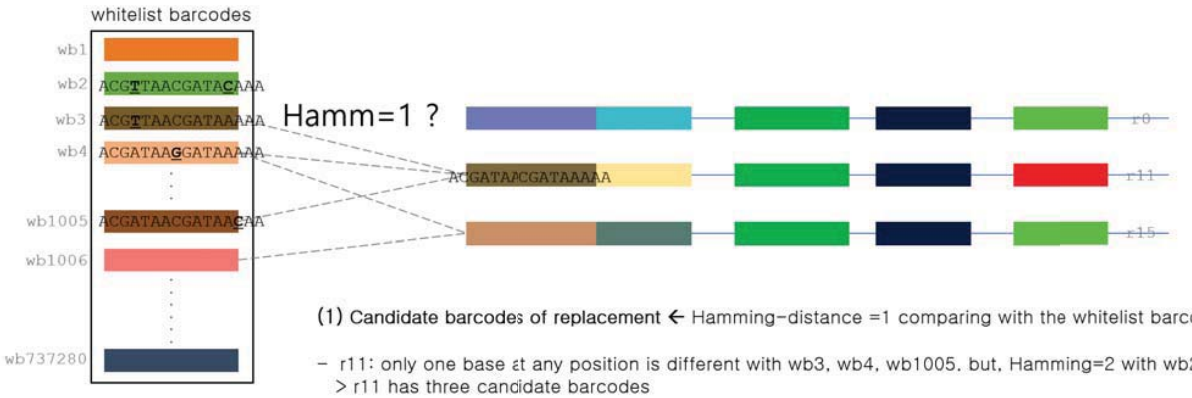
Identifying error-corrected barcode sequence



- All reads are perfectly matched with the whitelist barcodes except for r0, r11, r15
- Reads of r0, r11, r15 are tested for possibility of replacing barcodes, in the next step

25

Identifying error-corrected barcode sequence



(1) Candidate barcodes of replacement ← Hamming-distance =1 comparing with the whitelist barcodes

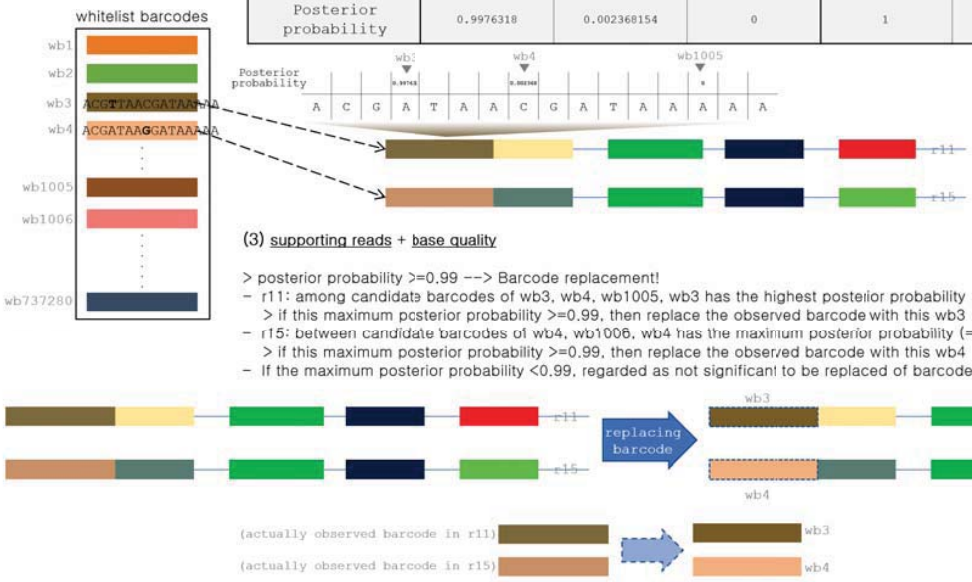
- r11: only one base at any position is different with wb3, wb4, wb1005, but, Hamming=2 with wb2
  > r11 has three candidate barcodes
- r15: two candidate barcodes of wb4, wb1006
- r0: No Hamm=1
- Estimate probabilities to be changeable or not with r11, r15 that have candidate barcodes, in the next step

26

Identifying error-corrected barcode sequence

	r11			r15	
candidate barcodes	wb3	wb4	wb1005	wb4	wb1006
Posterior probability	0.9976318	0.002368154	0	1	0

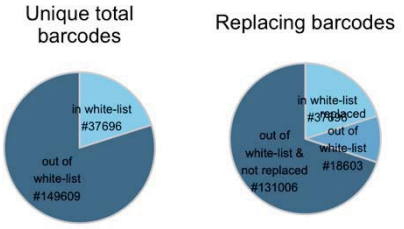
Statistically Reliable to be replaced?

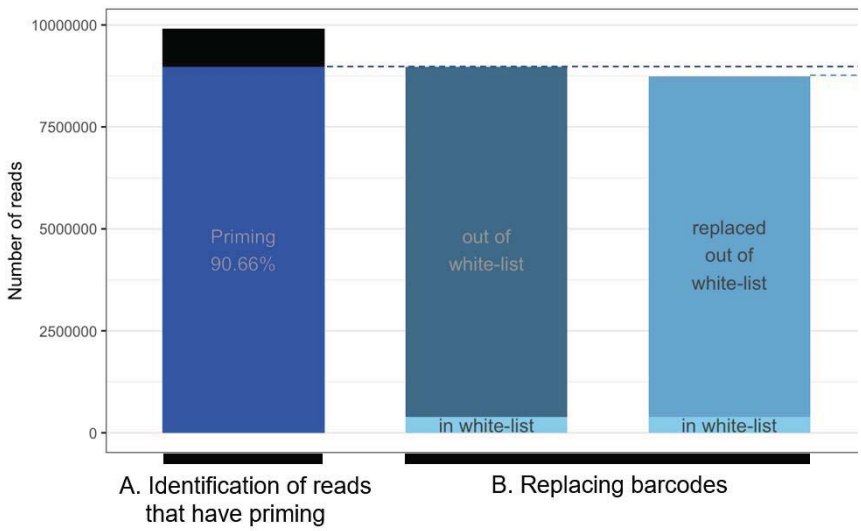


(3) supporting reads + base quality

> posterior probability >=0.99 --> Barcode replacement!
- r11: among candidate barcodes of wb3, wb4, wb1005, wb3 has the highest posterior probability (= 0.9976318)
  > if this maximum posterior probability >=0.99, then replace the observed barcode with this wb3
- r15: between candidate barcodes of wb4, wb1006, wb4 has the maximum posterior probability (= 1)
  > if this maximum posterior probability >=0.99, then replace the observed barcode with this wb4
- If the maximum posterior probability <0.99, regarded as not significant to be replaced of barcode and rescued of the read

Identifying error-corrected barcode sequence





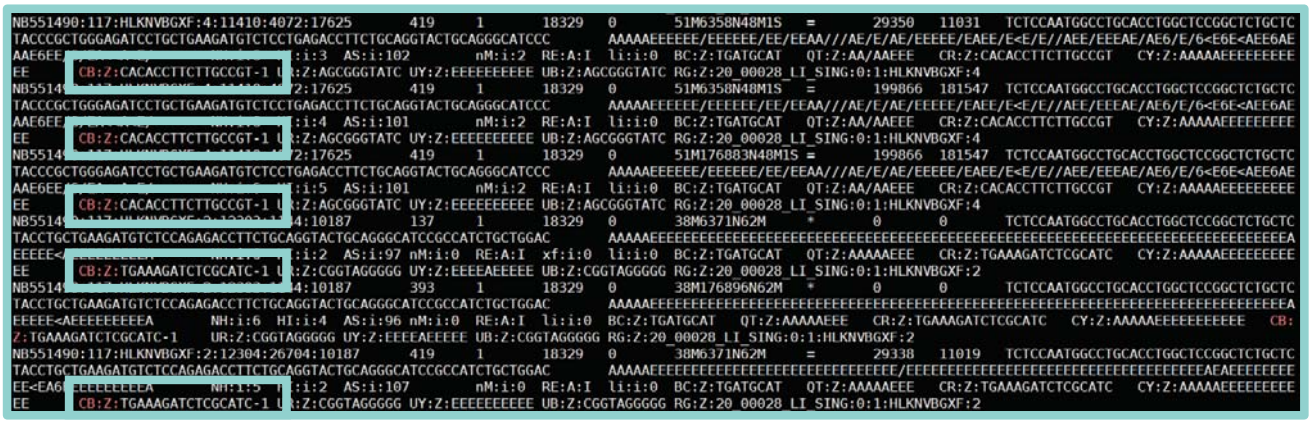
Output BAM

(White-listed barcodes)

- 10X v2.chemistry: `737K-august-2016.txt`
- 10X v3.chemistry: `3M-february-2018.txt`

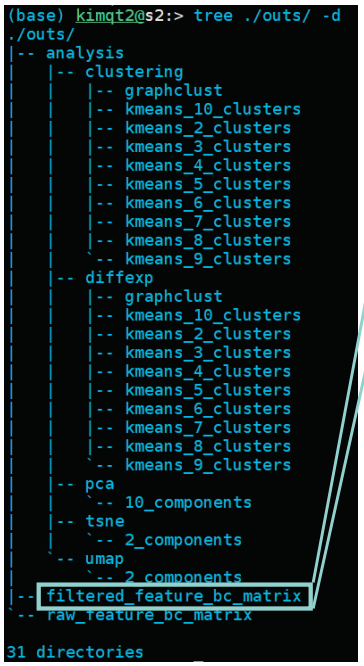
(Length of UMI)

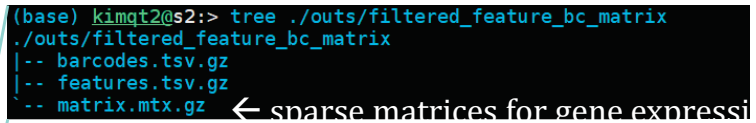
- 10X v2.chemistry: `10`
- 10X v3.chemistry: `12`

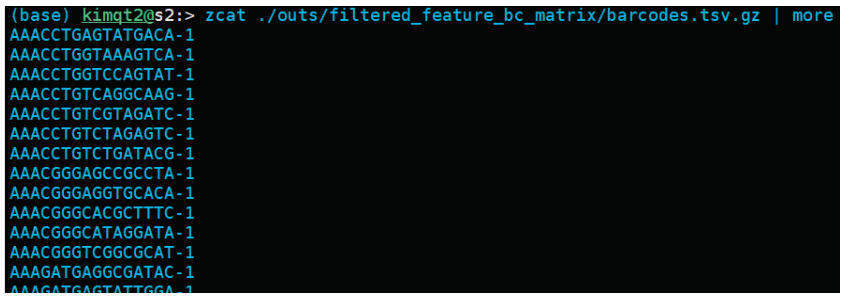


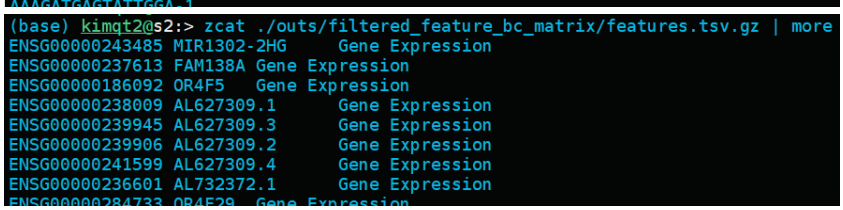
29

Output matrices





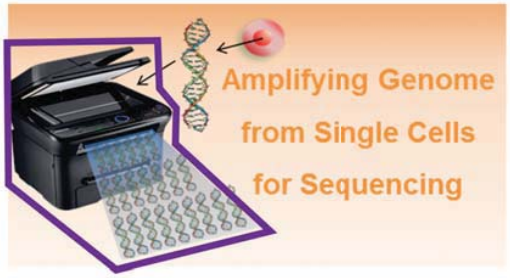


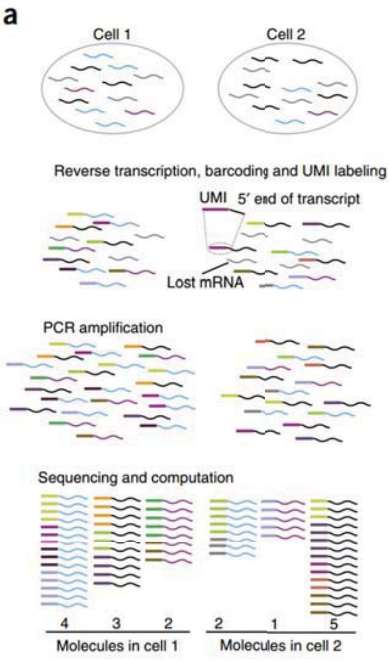


30

Estimation of relative level of gene expression & Normalization of their abundances

PCR-derived duplications





nature methods

Explore content ▾ About the journal ▾ Publish with us ▾ Subscribe

nature > nature methods > brief communications > article

Published: 22 December 2013

Quantitative single-cell RNA-seq with unique molecular identifiers

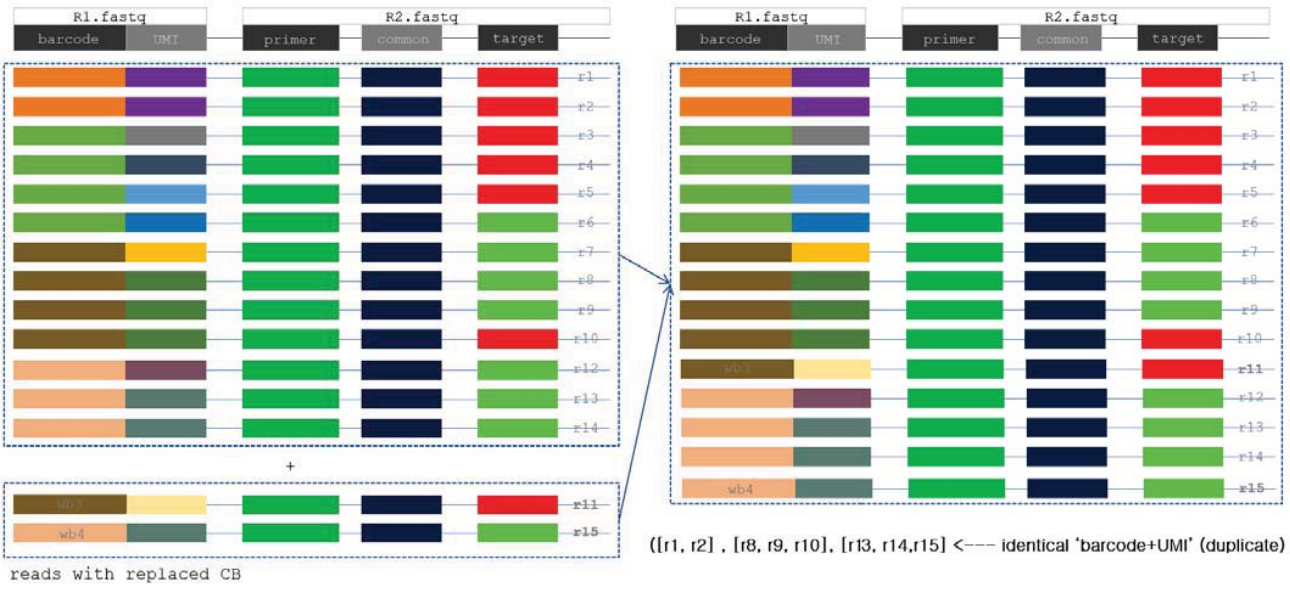
Saiful Islam, Amit Zeisel, Simon Joost, Gioele La Manno, Pawel Zajac, Maria Kasper, Peter Lönnerberg & Sten Linnarsson

Nature Methods **11**, 163–166 (2014) | Cite this article

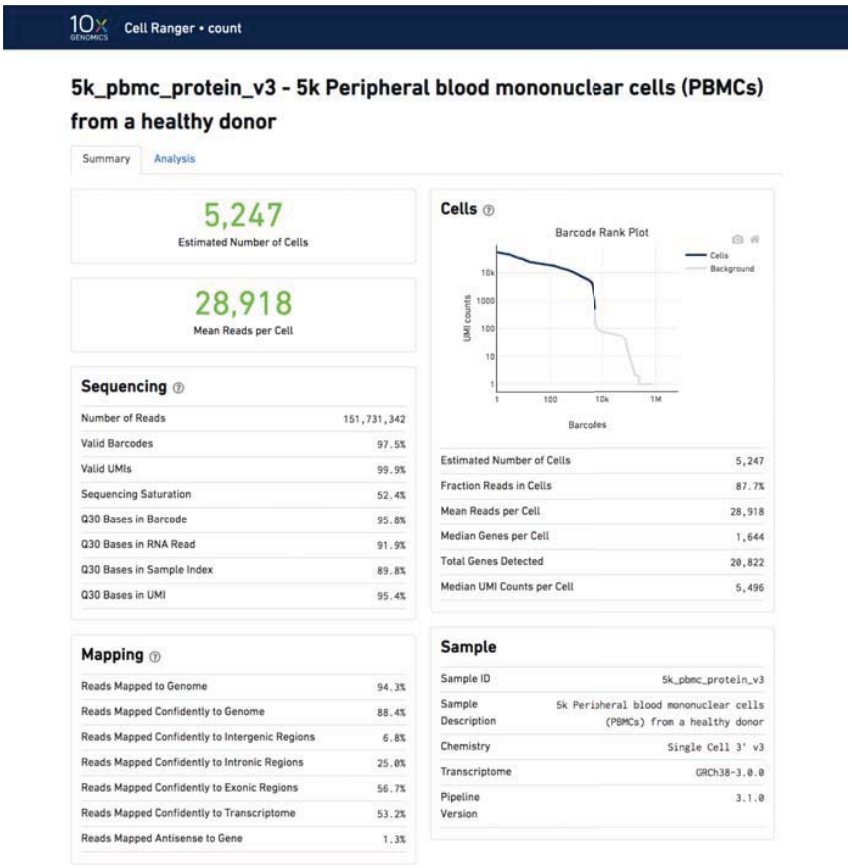
60k Accesses | **622** Citations | **42** Altmetric | Metrics

duplication을 구별하기 위해서 Unique sequence tagging: UMI (Unique Molecular Identifier)

Estimation of relative level of gene expression & Normalization of their abundances



Output matrices



10x GENOMICS Cell Ranger • count

5k_pbmc_protein_v3 - 5k Peripheral blood mononuclear cells (PBMCs) from a healthy donor

Summary | Analysis

5,247
Estimated Number of Cells

28,918
Mean Reads per Cell

Sequencing

Number of Reads	151,731,342
Valid Barcodes	97.5%
Valid UMIs	99.9%
Sequencing Saturation	52.4%
Q30 Bases in Barcode	95.8%
Q30 Bases in RNA Read	91.9%
Q30 Bases in Sample Index	89.8%
Q30 Bases in UMI	95.4%

Mapping

Reads Mapped to Genome	94.3%
Reads Mapped Confidently to Genome	88.4%
Reads Mapped Confidently to Intergenic Regions	6.8%
Reads Mapped Confidently to Intronic Regions	25.0%
Reads Mapped Confidently to Exonic Regions	56.7%
Reads Mapped Confidently to Transcriptome	53.2%
Reads Mapped Antisense to Gene	1.3%

Cells

Barcode Rank Plot

Estimated Number of Cells	5,247
Fraction Reads in Cells	87.7%
Mean Reads per Cell	28,918
Median Genes per Cell	1,644
Total Genes Detected	20,822
Median UMI Counts per Cell	5,496

Sample

Sample ID	5k_pbmc_protein_v3
Sample Description	5k Peripheral blood mononuclear cells (PBMCs) from a healthy donor
Chemistry	Single Cell 3' v3
Transcriptome	GRCh38-3.0.0
Pipeline Version	3.1.0

33

Seurat R package:
the most popular tool package processing 10X output data

https://satijalab.org/seurat

1) Read the 10X output data
2) QC and select cells for further analysis
3) Normalize the data
4) Detection of variable genes across the single cells
5) Scale the data and remove unwanted sources of variation
6) Perform linear dimensional reduction
7) Determine statistically significant principal components
8) Cluster the cells
9) Run non-linear dimensional reduction
10) Find differentially expressed genes (cluster biomarkers)
11) Assign cell type identity to clusters
12) Further sub-dissect within cell types

34

Data loading (practice)

```
./cellrangers/
├── ctl.1
│   ├── barcodes.tsv.gz
│   ├── features.tsv.gz
│   └── matrix.mtx.gz
├── ctl.2
│   ├── barcodes.tsv.gz
│   ├── features.tsv.gz
│   └── matrix.mtx.gz
├── ctl.3
│   ├── barcodes.tsv.gz
│   ├── features.tsv.gz
│   └── matrix.mtx.gz
├── luad.1
│   ├── barcodes.tsv.gz
│   ├── features.tsv.gz
│   └── matrix.mtx.gz
├── luad.2
│   ├── barcodes.tsv.gz
│   ├── features.tsv.gz
│   └── matrix.mtx.gz
└── luad.3
    ├── barcodes.tsv.gz
    ├── features.tsv.gz
    └── matrix.mtx.gz

6 directories, 18 files
```

Data loading

```
home = "D:/GoogleDrive/Documents/Lectures/2024.1st/2024KSBi_BIML/data4practice" ;
setwd(home) ;
library(Seurat) ; library(ggplot2) ;

# where CellRanger outputs
cellrangers = dir(paste0(getwd(),"/CellRangerOuts")) ;
cellrangers
# "ctl.1"  "ctl.2"  "ctl.3"  "luad.1" "luad.2" "luad.3"

# load each CellRanger output and merge as a seurat.object
for (i in 1:length(cellrangers)){
  data.i = Read10X(data.dir = paste0(getwd(),"/CellRangerOuts/",cellrangers[i])) ;

  colnames(data.i) = paste0(cellrangers[i],".",colnames(data.i)) ;
  obj.i = CreateSeuratObject(counts= data.i, project="lung_obj", min.cells=3, min.features=300) ;
  obj.i$orig.ident = cellrangers[i] ;
  obj.i[["percent.mt"]] = PercentageFeatureSet(obj.i, "^MT-") ;

  cat(paste0("i = ",i," | ",cellrangers[i],"\n")) ;
  if(i==1){luadobj = obj.i} else {luadobj = merge(luadobj, obj.i)}
} ;
save(luadobj, file=paste0(home,"luadobj.rda")) ;
```

36

Data loading

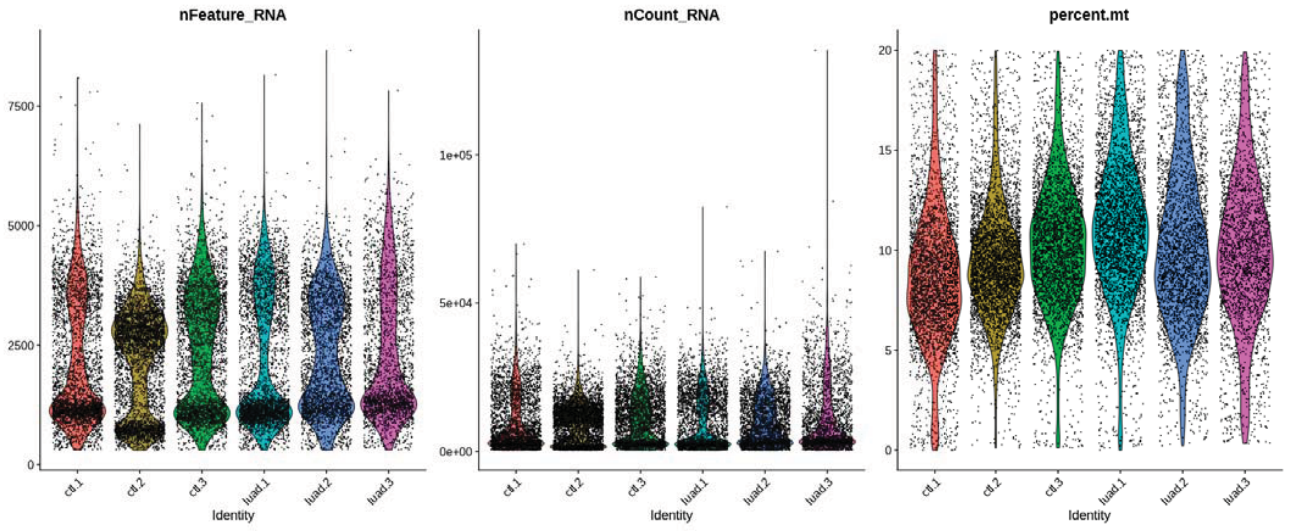
```
> luadobj
An object of class Seurat
20776 features across 21165 samples within 1 assay
Active assay: RNA (20776 features, 0 variable features)
>
> head(luadobj@meta.data)
                       orig.ident nCount_RNA nFeature_RNA percent.mt
ctl.1.AAACCCAGTTATGACC      ctl.1       6342         1947   9.350363
ctl.1.AAACCCAGTTCGAGCC      ctl.1       2255         1046   5.986696
ctl.1.AAACGAACAAGGCGTA      ctl.1      31132         4264  10.741359
ctl.1.AAACGAACATCTTCGC      ctl.1       3025         1153   7.206612
ctl.1.AAACGAAGTGCGTTTA      ctl.1       2186         1211   6.953339
ctl.1.AAACGAAGTTGGGCCT      ctl.1       2677         1176  10.646246
>
>
> table(luadobj@meta.data$orig.ident)

 ctl.1  ctl.2  ctl.3 luad.1 luad.2 luad.3
  3565   4511   3656   3670   3091   2672
>
```

37

Filtering out poor quality cells

```
luadobj = subset(luadobj, subset = nFeature_RNA > 200 & percent.mt < 20) ;
```

```
Idents(luadobj) = "orig.ident"

VlnPlot(luadobj, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
```



38

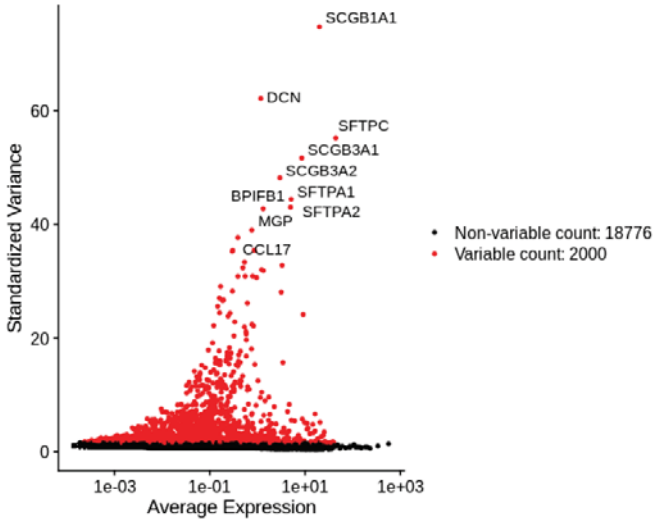
Normalize expression matrix and identify top variable genes

```
> # normalize expression matrix and identify top 2000 varialbe genes
> luadobj <- NormalizeData(luadobj, normalization.method = "LogNormalize", scale.factor = 10000)
Performing log-normalization
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
>
> luadobj <- FindVariableFeatures(luadobj, selection.method = "vst", nfeatures = 2000)
Calculating gene variances
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Calculating feature variances of standardized and clipped values
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
```

39

Normalization

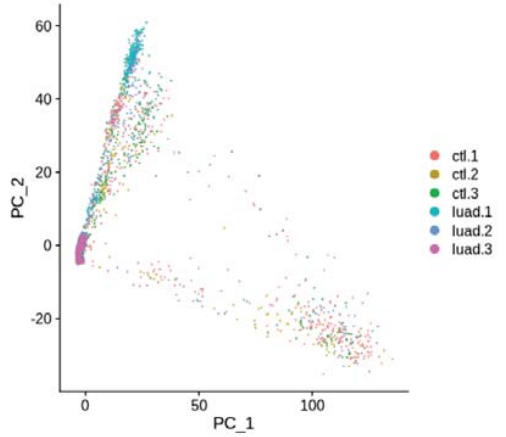
```
# Identify the 10 most highly variable genes
top10 <- head(VariableFeatures(luadobj), 10)

# plot variable features with and without labels
variable_plot <- VariableFeaturePlot(luadobj)
variable_plot.wTop10 <- LabelPoints(plot = variable_plot, points = top10, repel = TRUE)
variable_plot.wTop10
```

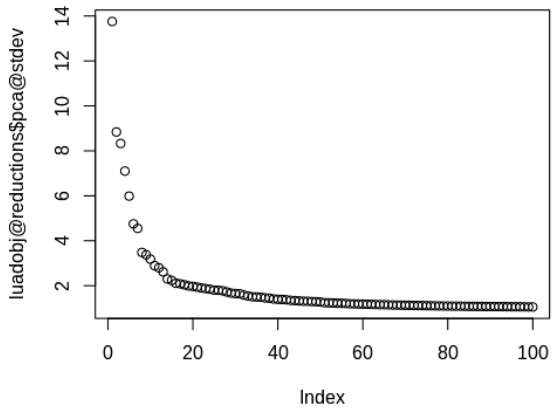


40

Normalization and PCA

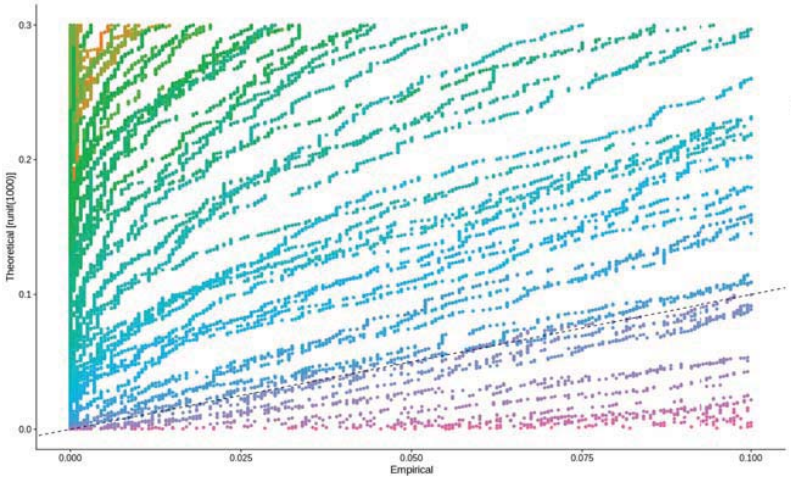
```
luadobj = ScaleData(luadobj, features=rownames(luadobj)) ;
## in case of removing unwanted sources of variation like MT contamination or cell cycling,
## regress out such heterogeneity
# luadobj = ScaleData(luadobj, features=rownames(luadobj), vars.to.regress="percent.mt") ;

luadobj = RunPCA(luadobj, features=VariableFeatures(object=luadobj), npcs=100) ;
DimPlot(luadobj, reduction = "pca")
plot(luadobj@reductions$pca@stdev)
```

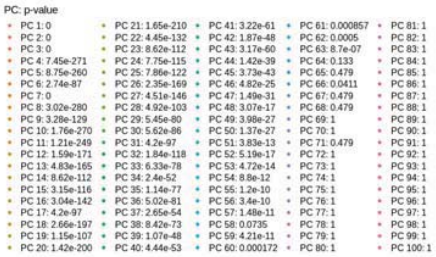




Estimate statistical significances of PCs

```
luadobj = JackStraw(luadobj, num.replicate=100, dims=100) ;
luadobj = ScoreJackStraw(luadobj, dims=1:100) ;
JackStrawPlot(luadobj, dims=1:100)
```

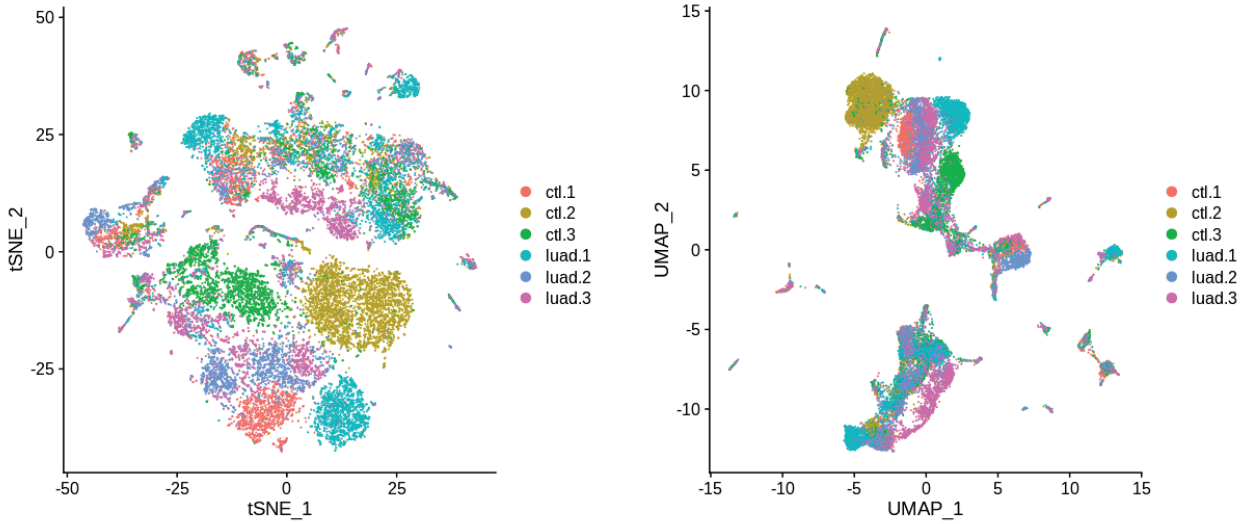




42

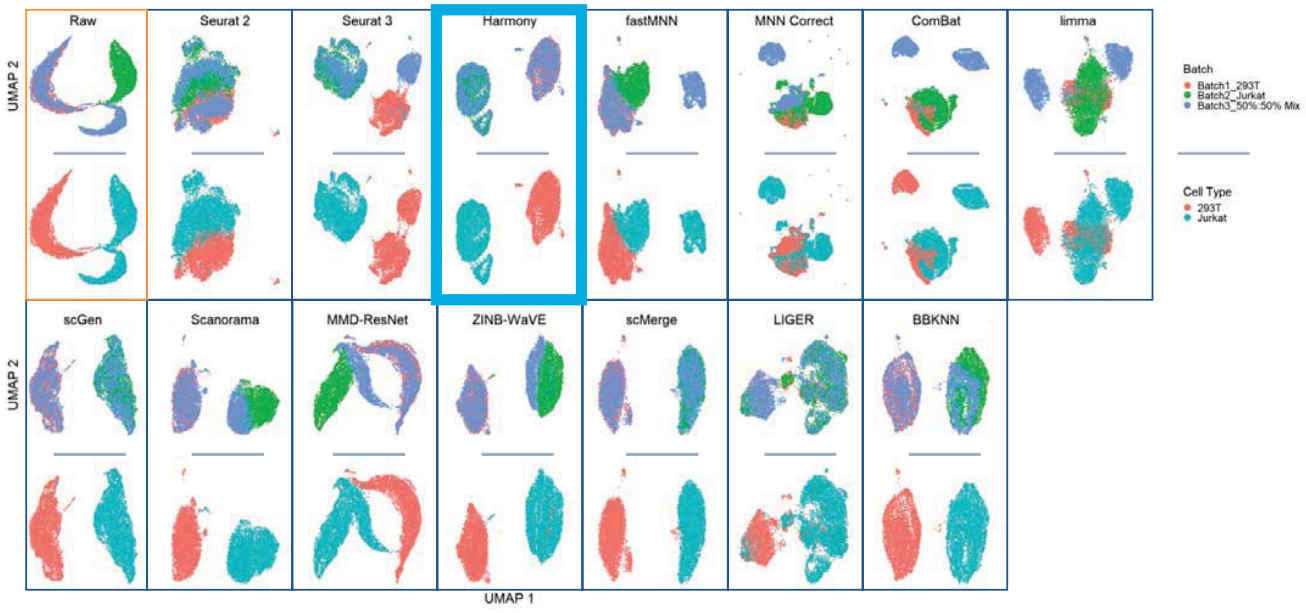
Dimensional reduction and visualization

```
luadobj = RunTSNE(luadobj, reduction="pca", dims=1:60) ;
luadobj = RunUMAP(luadobj, reduction="pca", dims=1:60) ;
DimPlot(luadobj, reduction = "tsne") ;
DimPlot(luadobj, reduction = "umap") ;
```



43

Batch-correction



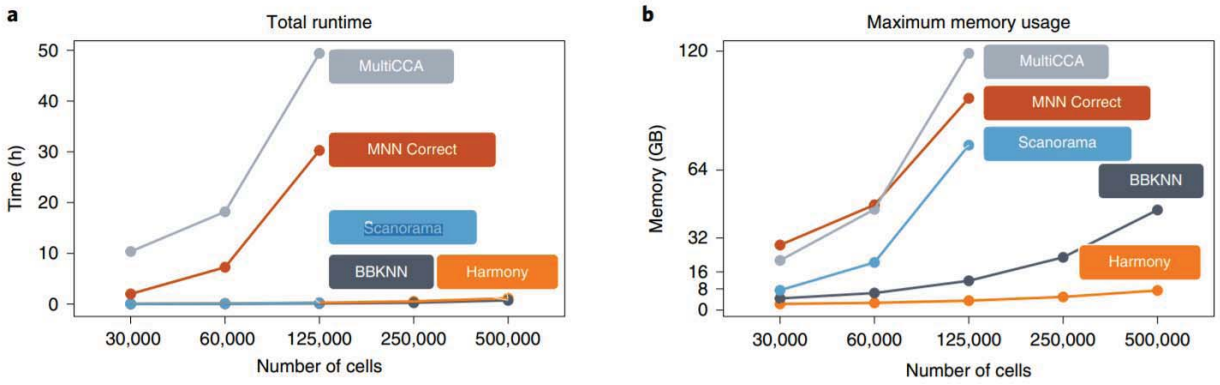
44

Batch-correction

nature methods ARTICLES
https://doi.org/10.1038/s41592-019-0619-0

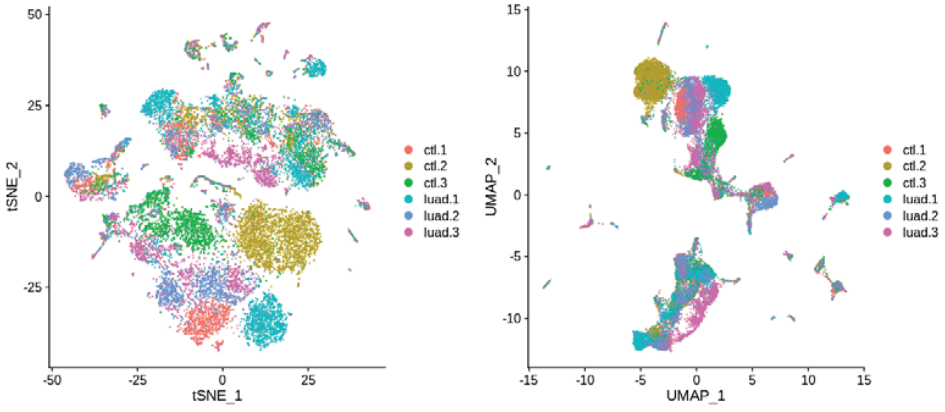
Fast, sensitive and accurate integration of single-cell data with Harmony

Ilya Korsunsky[1,2,3,4], Nghia Millard[1,2,3,4], Jean Fan[5], Kamil Slowikowski[1,2,3,4], Fan Zhang[1,2,3,4], Kevin Wei[2], Yuriy Baglaenko[1,2,3,4], Michael Brenner[2], Po-ru Loh[1,3,4] and Soumya Raychaudhuri[1,2,3,4,6*]



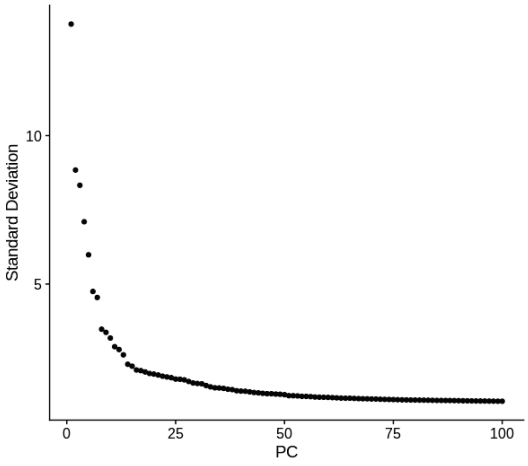
Batch-correction

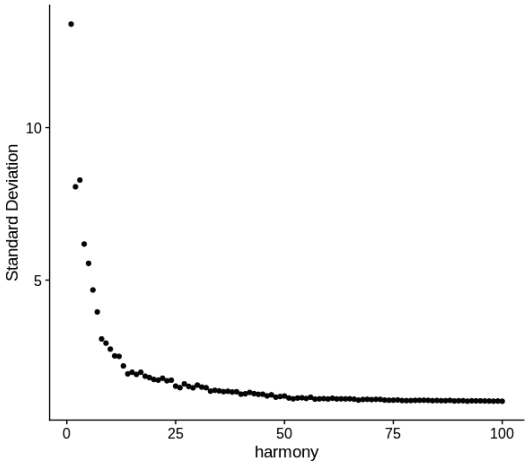
```
> head(luadobj@meta.data)
                    orig.ident nCount_RNA nFeature_RNA percent.mt
ctl.1.AAACCCAGTTATGACC      ctl.1       6342         1947   9.350363
ctl.1.AAACCCAGTTCGAGCC      ctl.1       2255         1046   5.986696
ctl.1.AAACGAACAAGGCGTA      ctl.1      31132         4264  10.741359
ctl.1.AAACGAACATCTTCGC      ctl.1       3025         1153   7.206612
ctl.1.AAACGAAGTGCGTTTA      ctl.1       2186         1211   6.953339
ctl.1.AAACGAAGTTGGGCCT      ctl.1       2677         1176  10.646246
> table(luadobj@meta.data$orig.ident)

 ctl.1  ctl.2  ctl.3 luad.1 luad.2 luad.3
  3565   4511   3656   3670   3091   2672
> unique(luadobj@meta.data$orig.ident)
[1] "ctl.1"  "ctl.2"  "ctl.3"  "luad.1" "luad.2" "luad.3"
```



Batch-correction

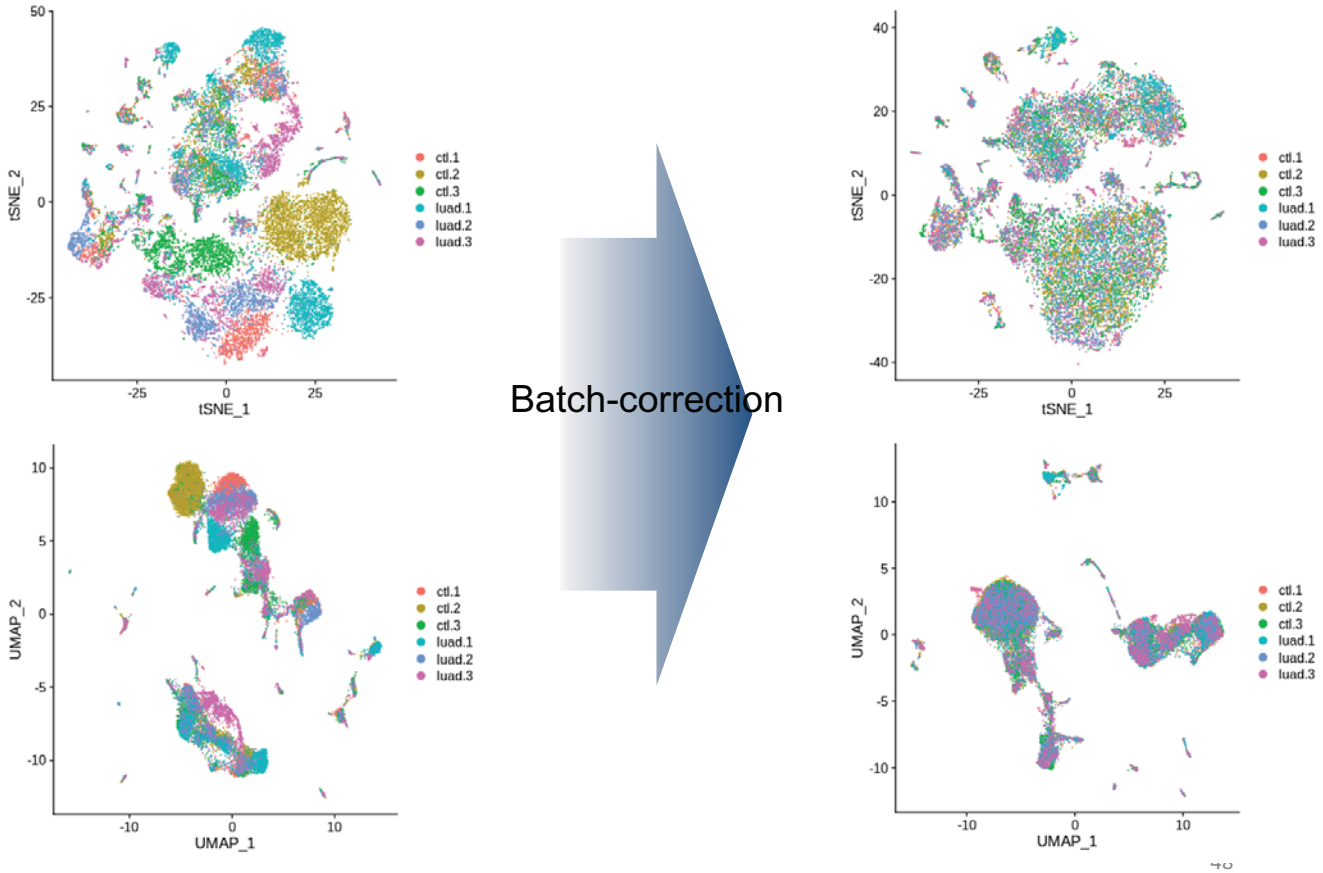
```
library(harmony) ;
luadobj = RunHarmony(luadobj, group.by.vars="orig.ident") ;
ElbowPlot(luadobj, reduction="harmony", ndims=100) ;

luadobj = RunTSNE(luadobj, reduction="harmony", dims=1:60, seed.use=1234) ;
luadobj = RunUMAP(luadobj, reduction="harmony", dims=1:60, seed.use=1234) ;
DimPlot(luadobj, reduction = "tsne") ;
DimPlot(luadobj, reduction = "umap") ;

save(luadobj, file="luadobj.rda")
```





47

Batch-correction



https://drive.google.com/drive/folders/1_qAYJFtitBjacAuLjzCictn0KIAN93G6?usp=sharing

Thank you!

KIMQTAE@ajou.ac.kr

KSBI | 한국생명정보학회

KSBi-BIML 2024

Single-cell RNA-sequencing analysis: Assignment of cell types (part2)

Kyu-Tae Kim
Ajou University School of Medicine

본 교육의 목표와 특징

단일세포 전사체 데이터 세포 종류 결정하기

- 클러스터링 분석의 의미를 이해한다.
- 클러스터링 종류와 방법을 이해한다.
- 세포 타입 결정 과정을 이해한다.
- 단일세포 전사체 데이터 clustering 과정을 이해한다.
- 단일세포 전사체로부터 cell type assignment 과정을 이해한다.

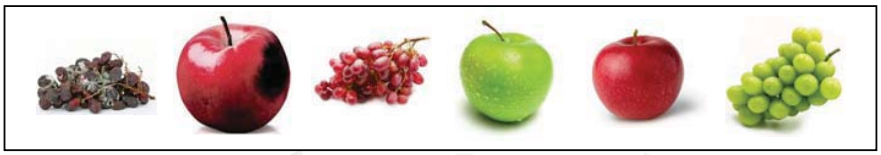
2

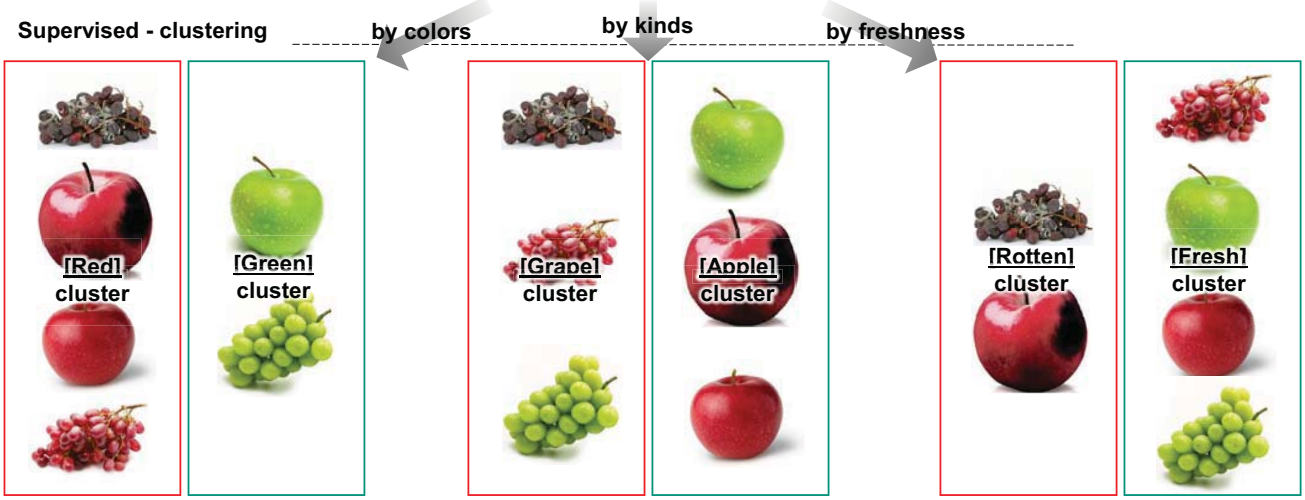
How to understand thousands of individual things?

Given that we have individual pieces of fruits (single-cell analysis), then how to sort these with which criteria? Color? Freshness? Kinds?

3

Clustering objects





4

Supervised vs. Un-supervised clustering

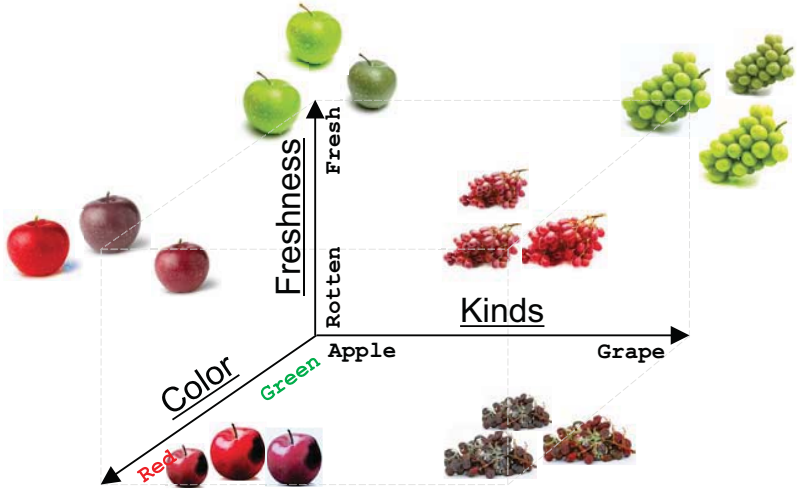
Supervised clustering	Un-supervised clustering
> The classes are predefined, and the task is to understand the basis for the classification from a set of labeled objects (training or learning set). > This information is then used to classify future observations. • Discriminant analysis • Class prediction • Supervised pattern recognition	> The classes are unknown a priori and need to be "discovered" from the data. • Cluster analysis • Class discovery • Unsupervised pattern recognition

5

Clustering analysis

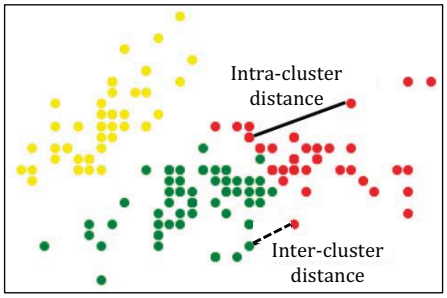
> **Finding groups** of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

> **Cluster**: a collection of 'similar' data

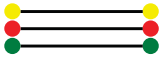


6

Evaluation of clustering

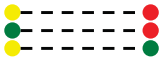


> Intra-cluster distance:

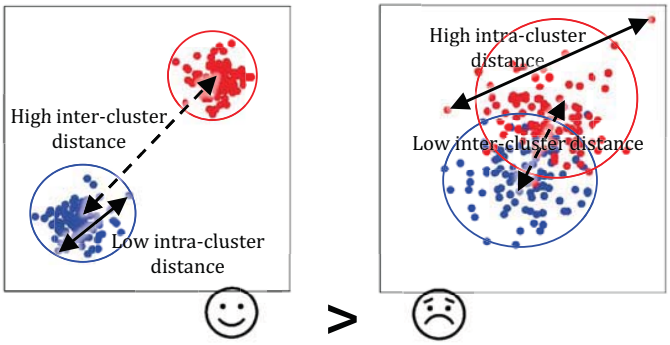


the distance among members of a cluster

> Inter-cluster distance:



the distance between two different clusters



> A **good clustering** method will produce high quality clusters with

Low intra-class distance = High intra-class similarity

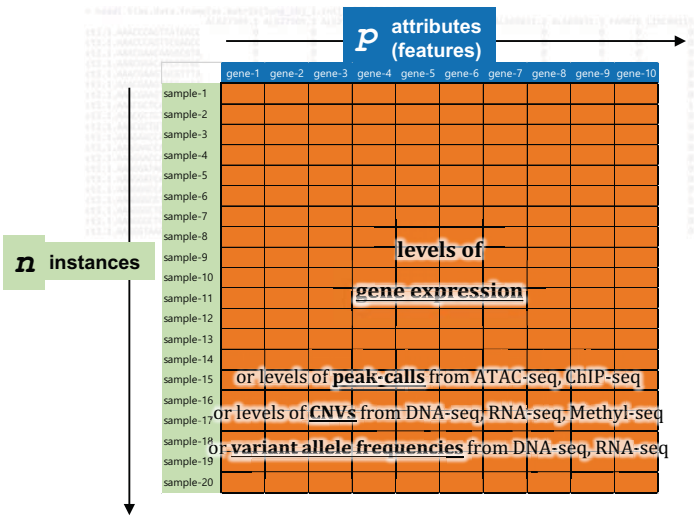
High inter-class distance = Low inter-class similarity

> How to determine **'similarity'**?

> How to measure **'distance'**?

클러스터링 종류와 방법

Similarity measures with a gene expression table



Data matrix

> n: size of the data (how many samples)

> p: attributes of the data (how many genes)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

Dimensionality

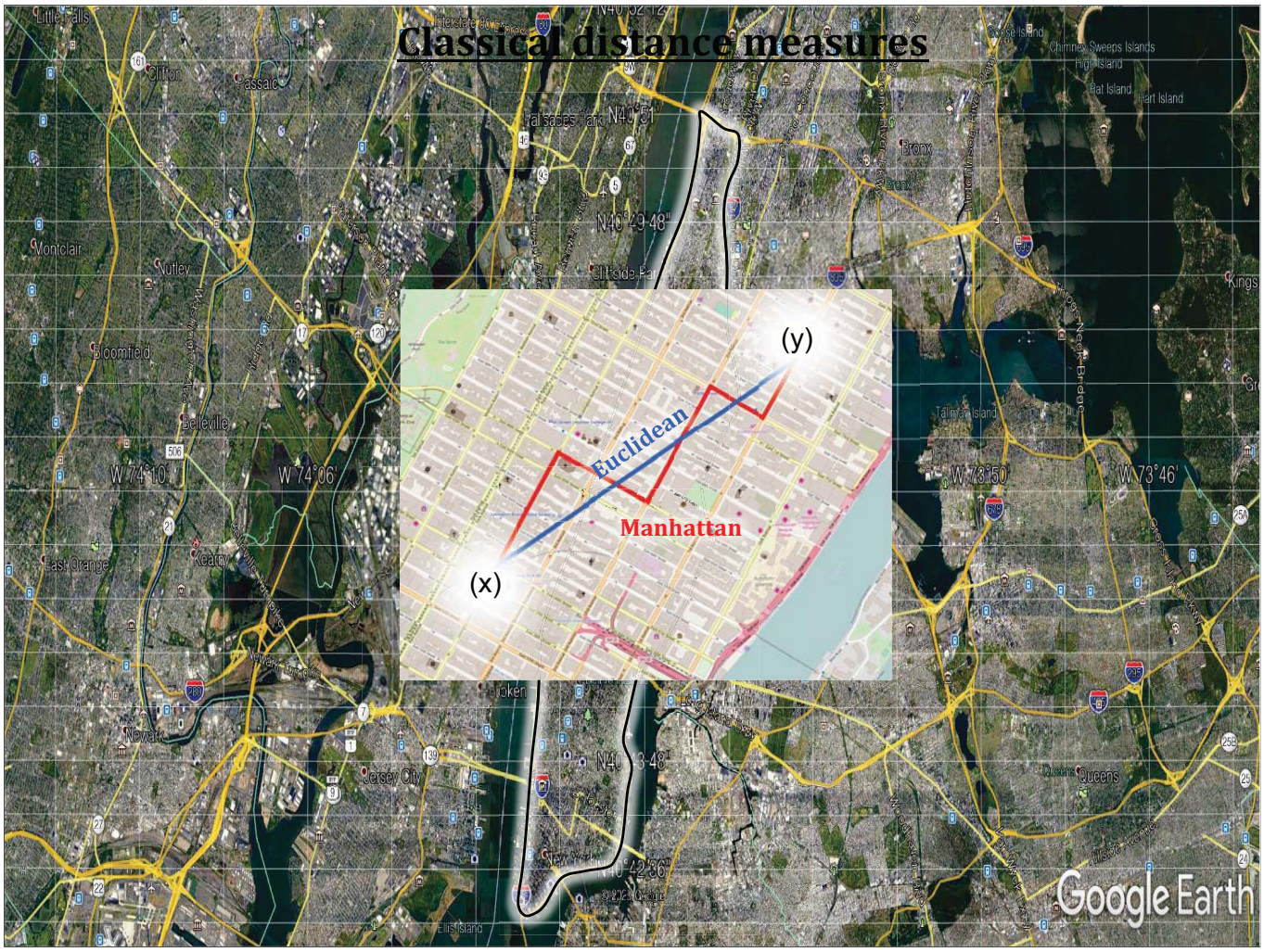
$= n \times p$

Distance matrix

(dissimilarity matrix)

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ \vdots & \vdots & \vdots & \\ d(n,1) & d(n,1) & \cdots \cdots & 0 \end{bmatrix}$$

9



Measures of relative distances

> Pearson correlation

- Measuring the degree of a linear relationship between two profiles

$$d_{cor}(x,y) = 1 - \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2 \sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

Parametric

> Eisen cosine correlation

- A special case of Pearson's correlation with ***x*** and ***y*** both replaced by zero

$$d_{eisen}(x,y) = 1 - \frac{\left|\sum_{i=1}^{n} x_i y_i\right|}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$$

> Spearman correlation

- Measuring the correlation between the rank of ***x*** and the rank of ***y*** variables

$$d_{spear}(x,y) = 1 - \frac{\sum_{i=1}^{n}(x'_i-\bar{x'})(y'_i-\bar{y'})}{\sqrt{\sum_{i=1}^{n}(x'_i-\bar{x'})^2 \sum_{i=1}^{n}(y'_i-\bar{y'})^2}}$$

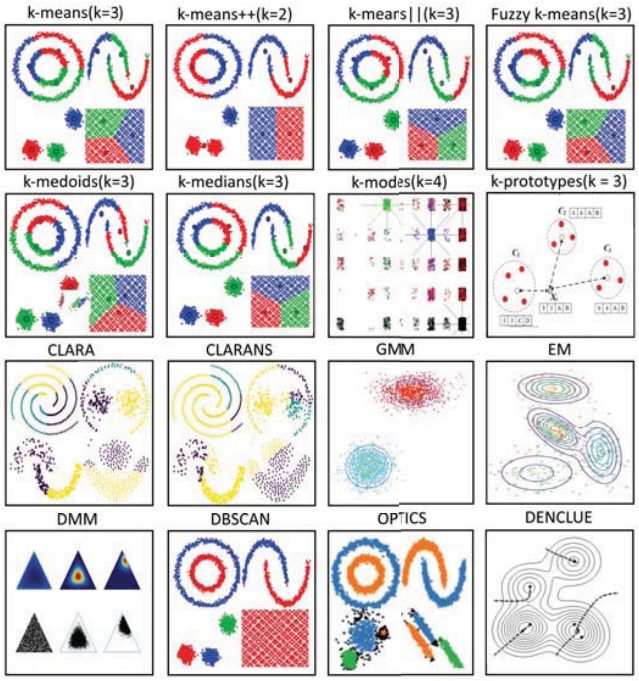
non-Parametric

> Kendall correlation

- Measuring the correspondence between the ranking of ***x*** and ***y*** variables

$$d_{kend}(x,y) = 1 - \frac{n_c - n_d}{\frac{1}{2}n(n-1)}$$

Clustering methods



> Hierarchical clustering

> Partitioning clustering

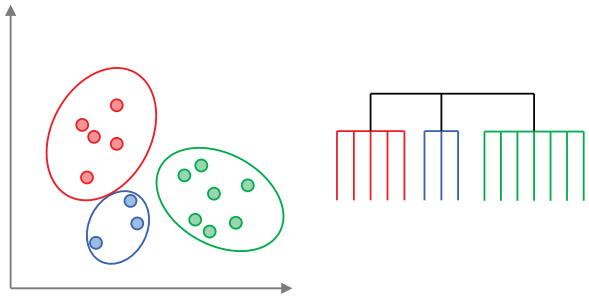
- K-medoids
- PAM (Partitioning Around Medoid)
- SOM (Self Organizing Maps)

> Advanced clustering

- Hybrid clustering methods
- Fuzzy clustering
- Model-based clustering
- Density-based clustering
- Graph-based clustering
- and ...

Hierarchical clustering

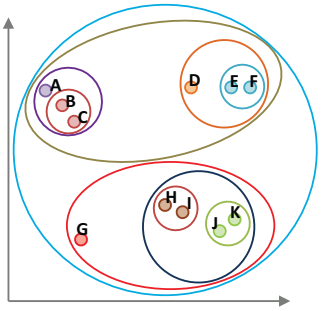
non-Hierarchical

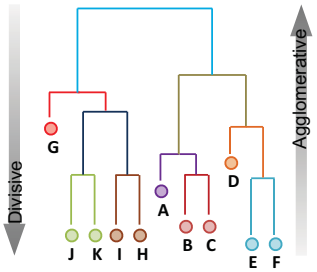


(Nested clusters)

(Dendrogram)

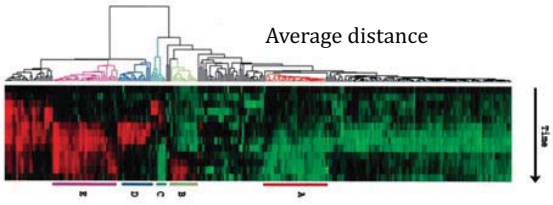
Hierarchical



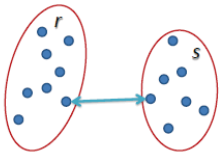


Hierarchical clustering

- Hierarchical clustering was the first algorithm used in microarray research to cluster genes. (David Bostein group, PNAS 1998)



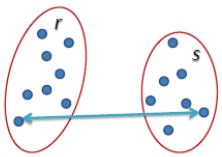
- First, each object is assigned to its own cluster. Then, iteratively, the two most similar clusters are joined, representing a new node of the clustering tree. The similarity matrix is updated. This process is repeated until only a single cluster remains. (agglomerative clustering)



$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

> Single linkage

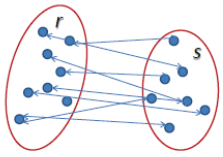
- Smallest distance



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

> Complete linkage

- Largest distance

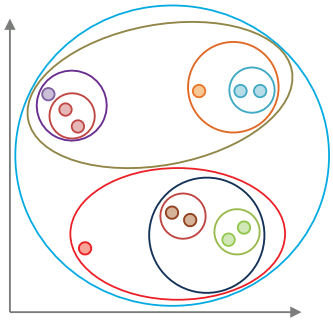


$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

> Average linkage

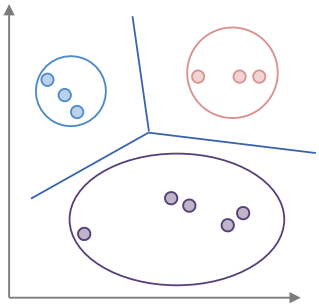
- Average distance

Partitioning clustering



> Hierarchical

- Clustering is hierarchical decomposition (i.e., multiple levels)
- It can not correct erroneous merges or splits



> Partitioning

- It find mutually exclusive clusters of spherical shape
- It may use mean or medoid to represent cluster center
- It may effective for small- to medium-size data sets

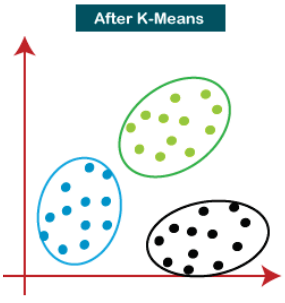
15

K-means clustering

- Number of cluster, ***K***, must be specified
- Each cluster is associated with an averaged point (centroid)
- Each point is assigned to the cluster with the closest centroid

- Basic algorithm

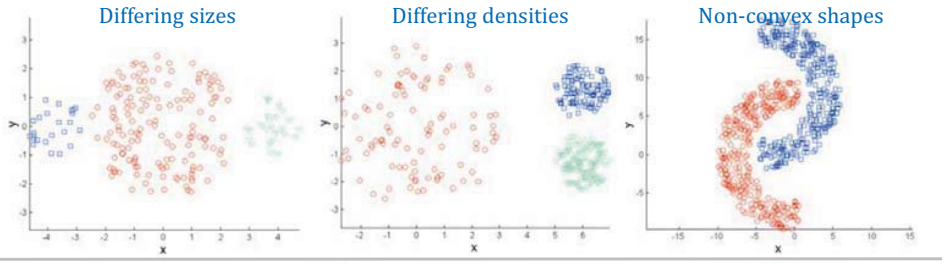
1: Select ***K*** points as the initial centroids.
2: **repeat**
3: From ***K*** clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids does not change



16

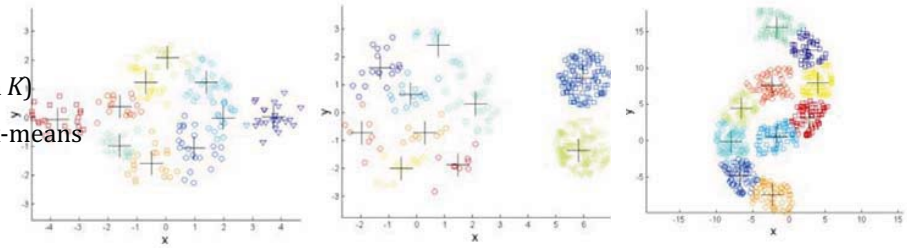
Limitation of *K-means* clustering

- Applicable only when mean is defined, then what about categorical data?
- Need to specify *K*, the number of clusters, in advance
- Unable to handle noisy data and outliers
- Not suitable to discover clusters with



Overcoming limitations

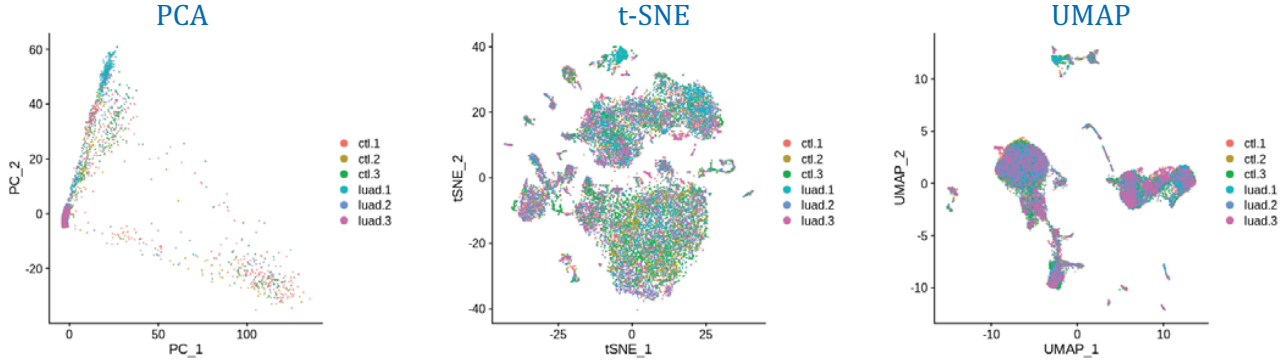
- Using many clusters (i.e., high *K*)
- Using K-medoids, instead of k-means which is sensitive to outliers



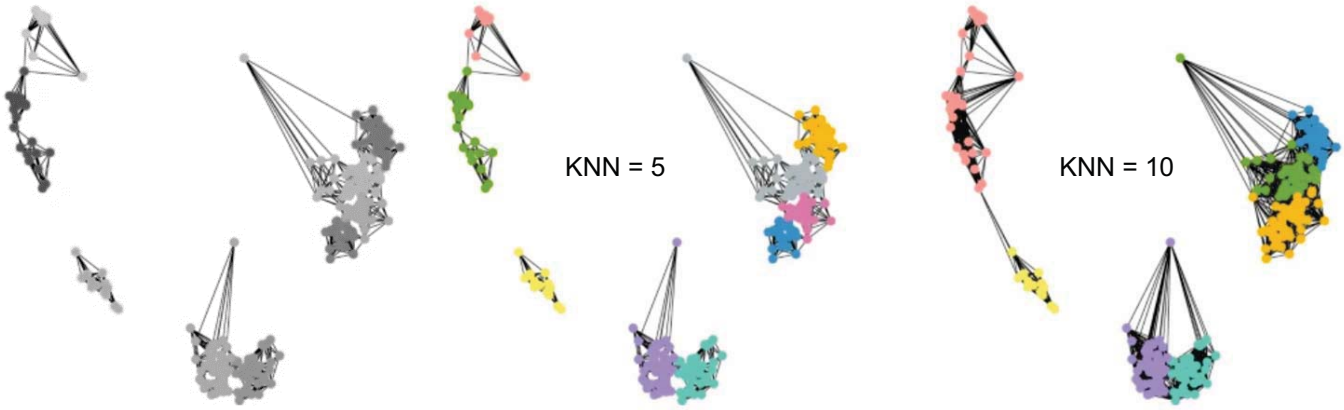
Dimensional reduction for visualization

> Projection methods

- PCA (Principal Component Analysis)
- t-SNE (t-distributed Stochastic Neighbor Embedding)
- UMAP (Uniform Manifold Approximation and Projection)



Graph-based clustering



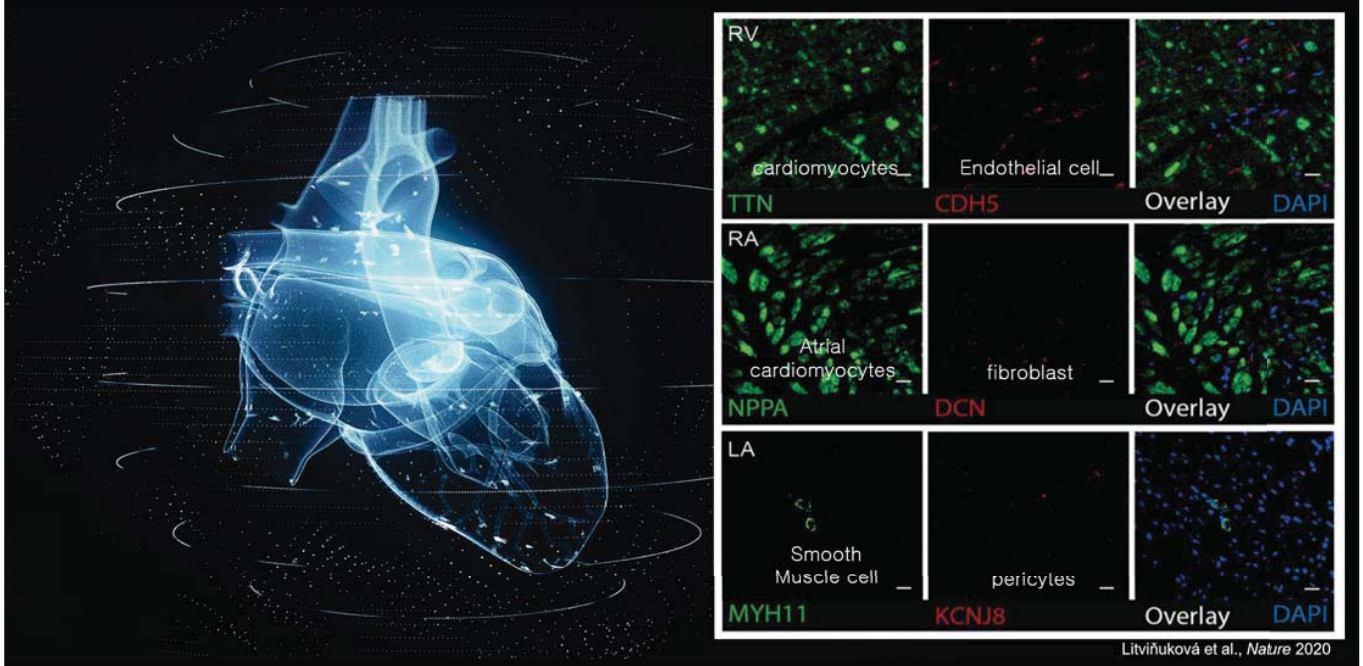
- Louvain community detection is applied to a shared-nearest-neighbor graph connecting the cells and finds tightly connected communities in the graph
- Increasing the number of neighbors when constructing the cell–cell graph indirectly decreases the resolution of graph-based clustering.

19

세포 유형 결정

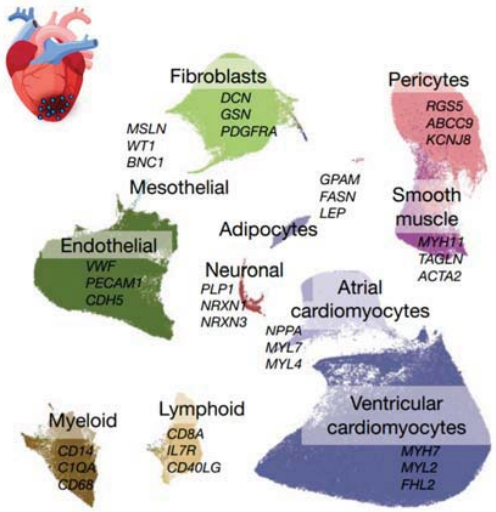
20

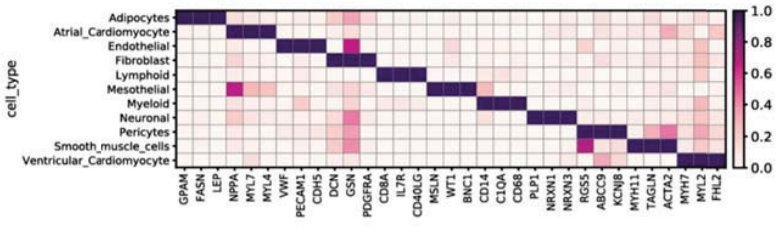
Cell-type assignment



Litviňuková et al., *Nature* 2020

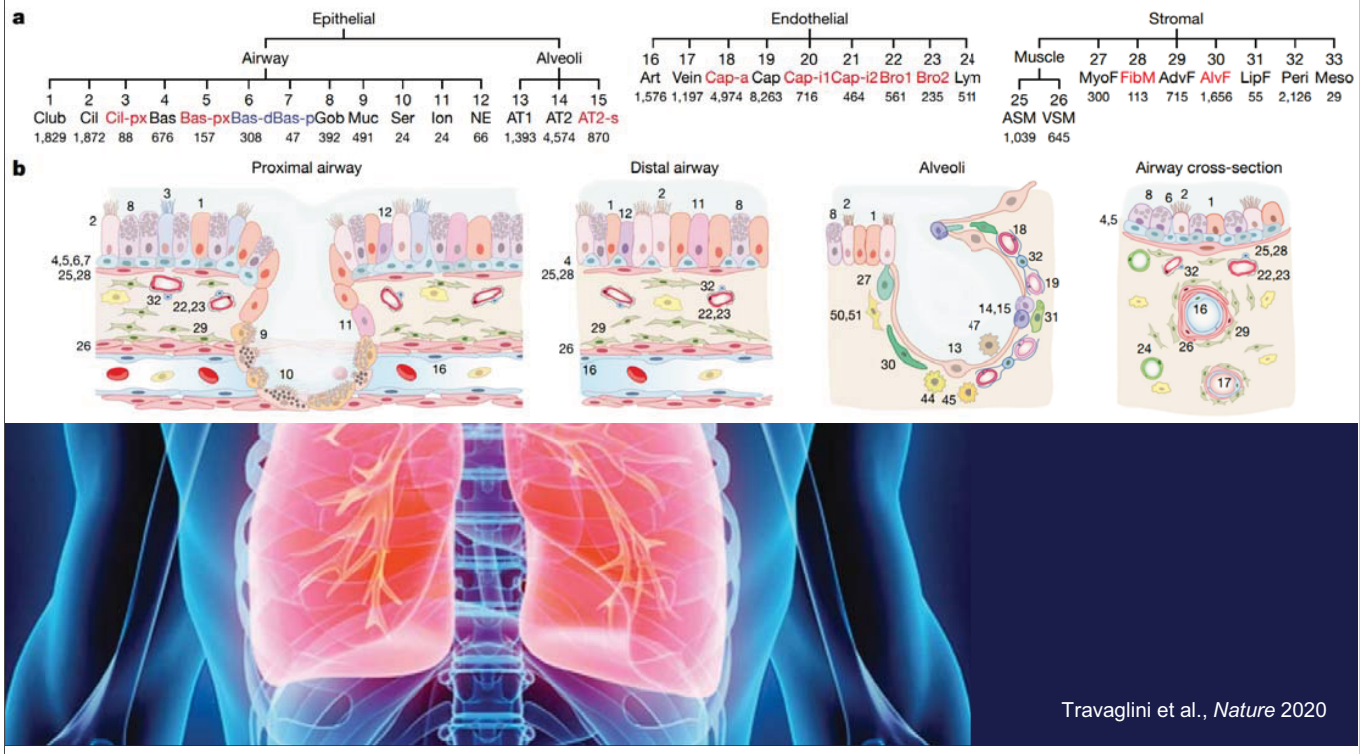
Cell-type assignment



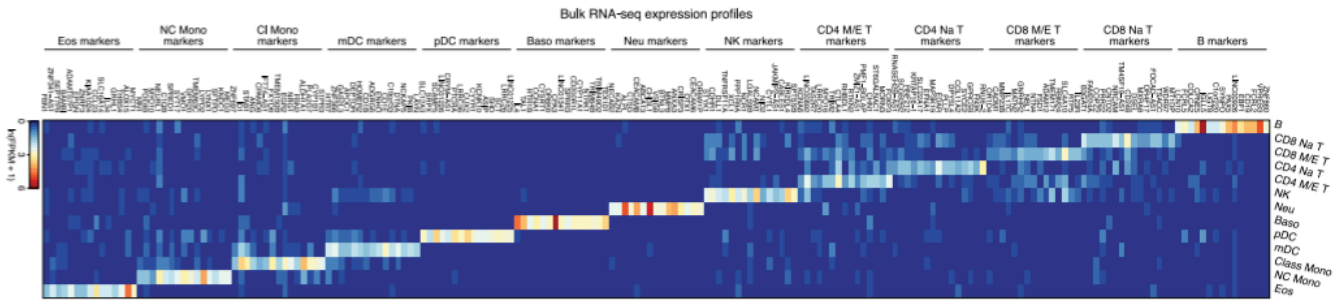


Litviňuková et al., *Nature* 2020

Cell-type assignment

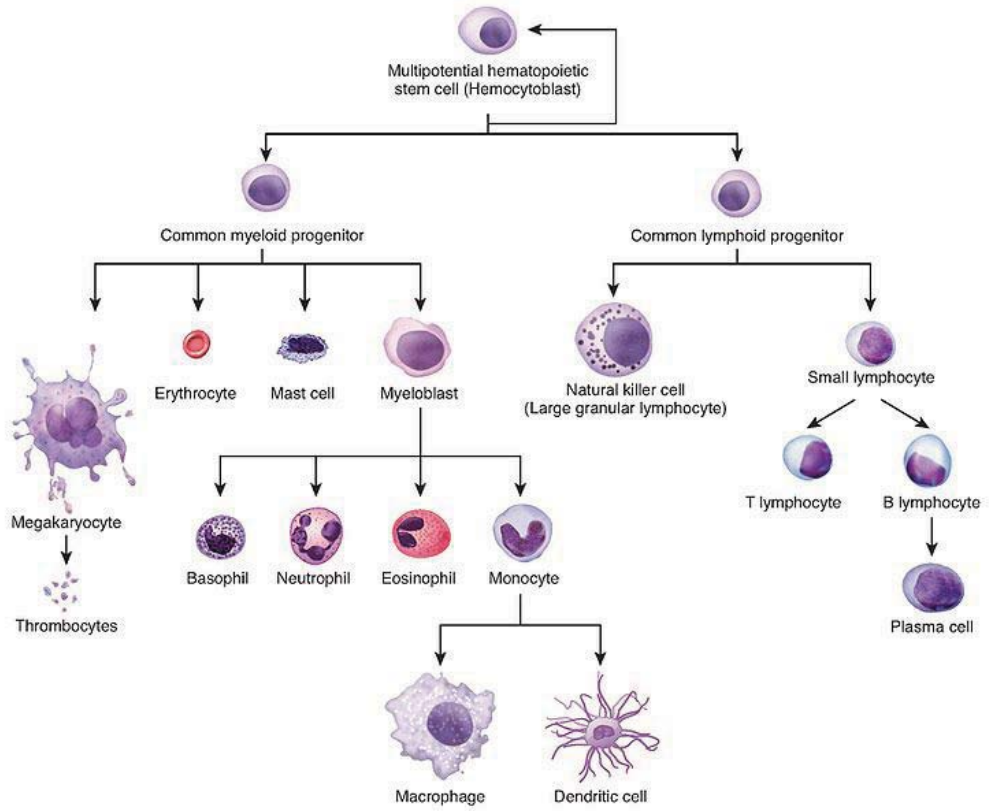


Cell-type assignment



Travaglini et al., *Nature* 2020

Hematopoiesis



25

Useful resources to identify cell type markers

[cell type gene expression] https://dice-database.org/

[cell type gene expression] http://biocc.hrbmu.edu.cn/CellMarker/index.jsp

[cell type gene expression] https://alona.panglaodb.se/

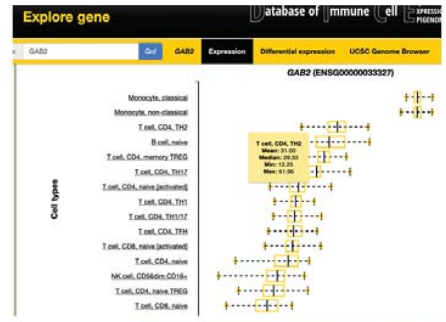
[cell type gene expression] https://cellxgene.cziscience.com/cellguide/

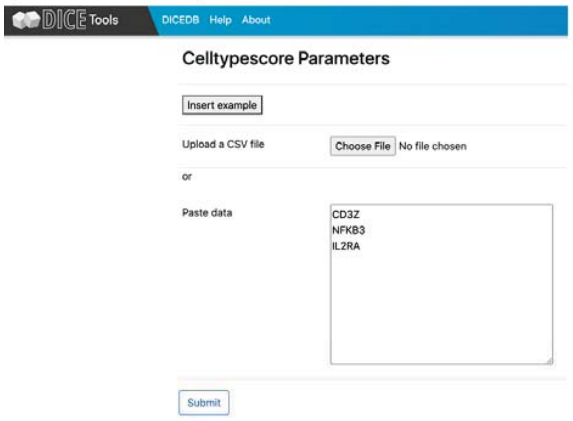
[cell type gene expression] https://www.celltypist.org/encyclopedia/Immune/v2

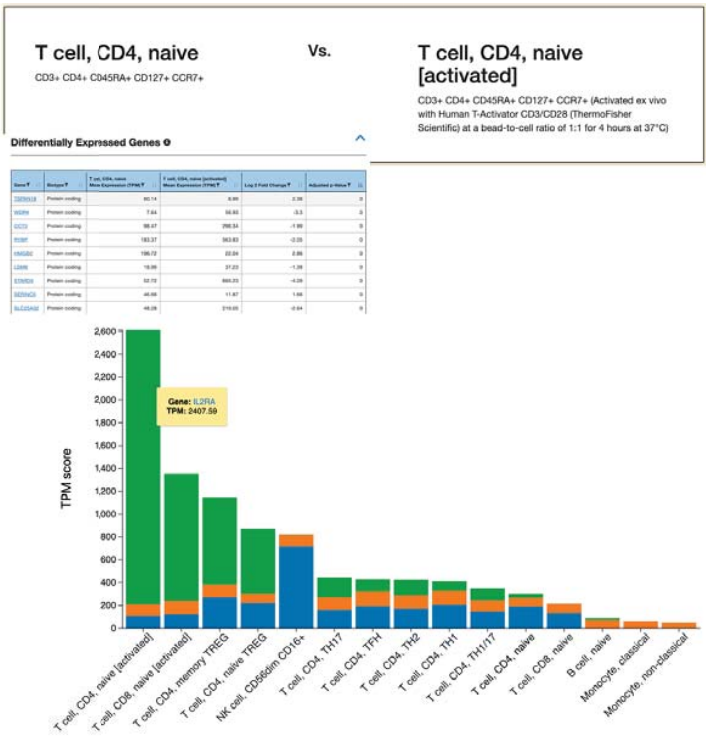
[cell types in blood/tissue marker ptn. expression] https://www.proteinatlas.org/

26

Useful resources to identify cell type markers

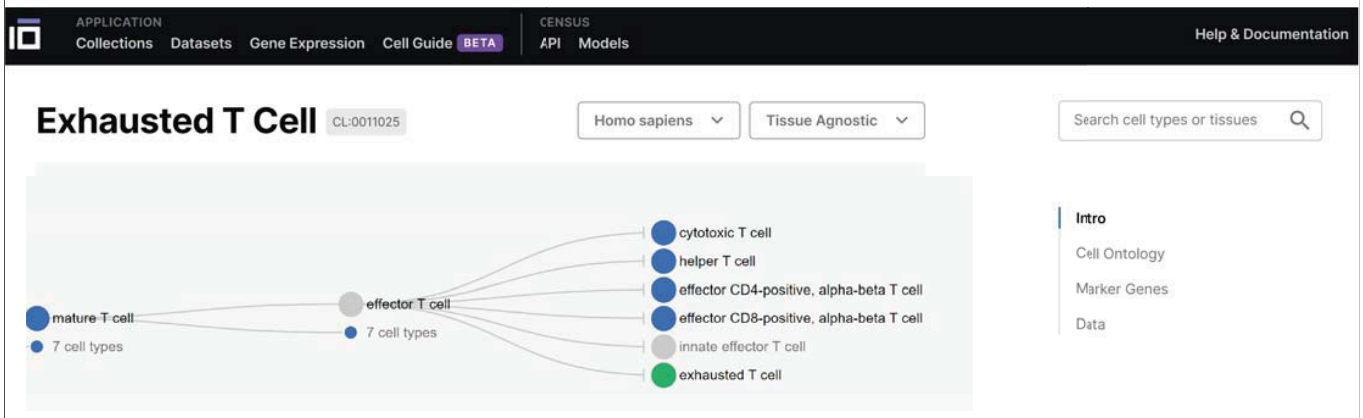






27

Useful resources to identify cell type markers



Marker Genes — Open in Gene Expression

Computational | Canonical

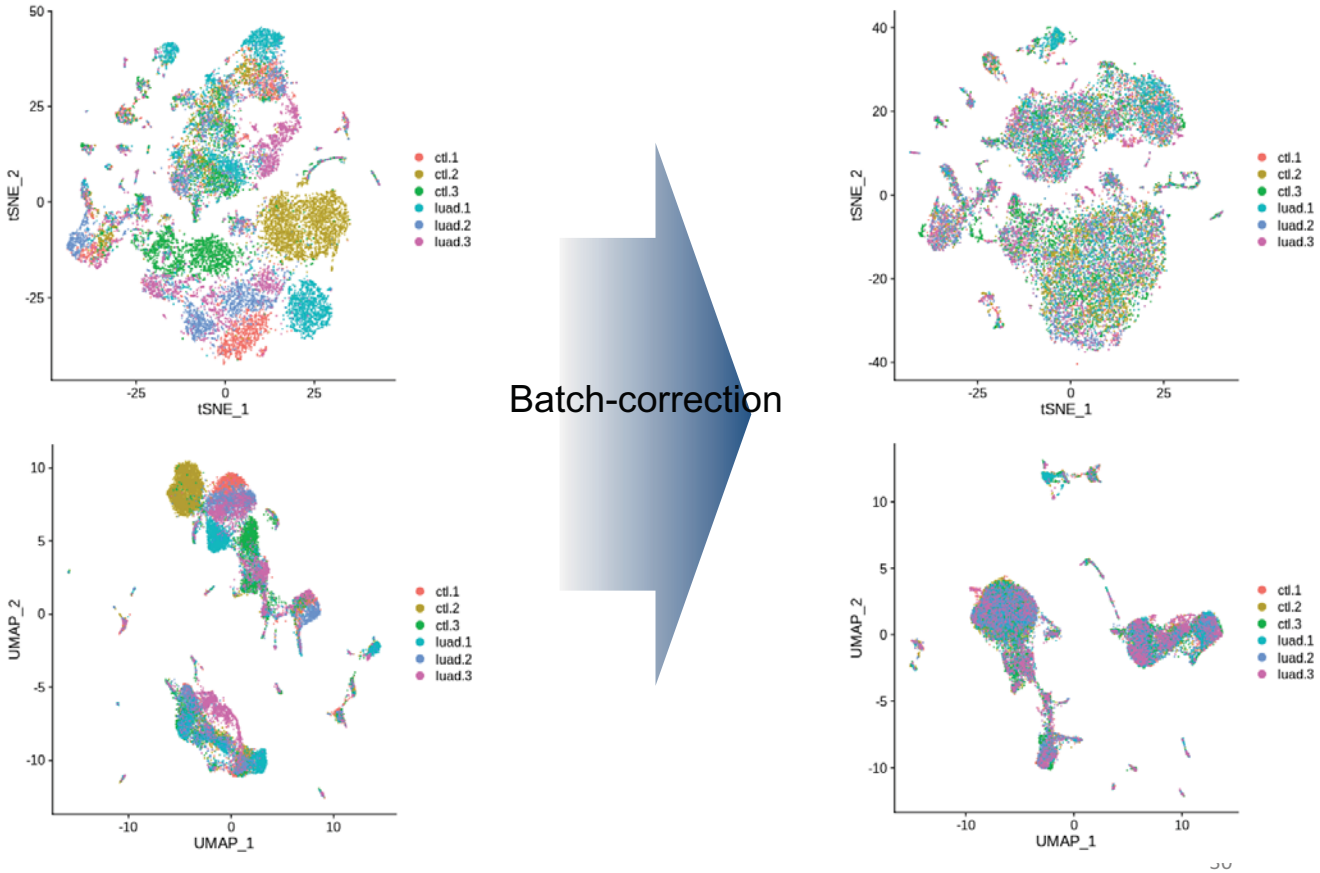
Symbol	Name	Effect Size	Specificity	Mean Expression	% of Cells
PTPRCAP	protein tyrosine phosphatase receptor type C associated protein	3.34	1.00	3.01	95.8
GZMK	granzyme K	3.14	1.00	3.23	93.8
CD3D	CD3 delta subunit of T-cell receptor complex	2.93	1.00	2.68	93.5
CST7	cystatin F	2.65	0.98	2.85	91.7
CD3E	CD3 epsilon subunit of T-cell receptor complex	2.61	0.99	2.73	94.8
CD27	CD27 molecule	2.09	1.00	2.48	84.7
NKG7	natural killer cell granule protein 7	2.06	0.94	3.39	85.9

28

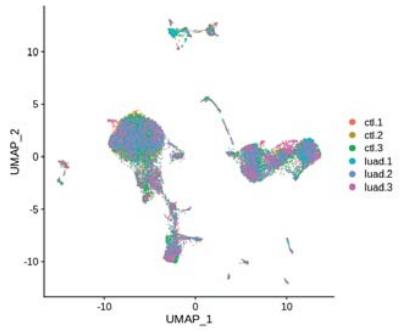
Data loading (practice)

https://drive.google.com/drive/folders/1_qAYJFtitBjacAuLjzCictn0KIAN93G6?usp=sharing

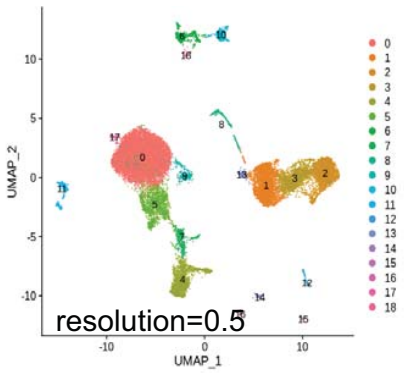
Batch-correction

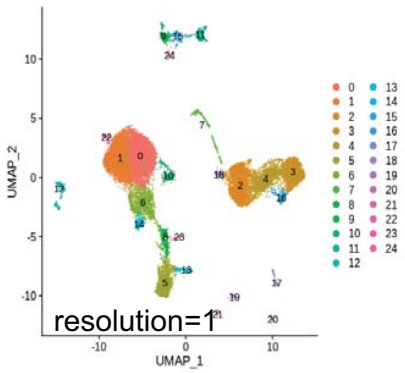


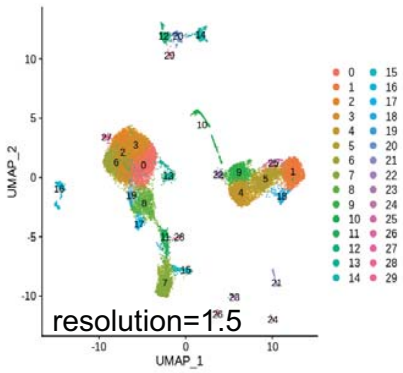
Clustering




```
> luadobj = FindNeighbors(luadobj, reduction="harmony", k.param=20, dims=1:60)
Computing nearest neighbor graph
Computing SNN
> luadobj = FindClusters(luadobj, resolution=1) ;
Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck

Number of nodes: 21165
Number of edges: 896920

Running Louvain algorithm...
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Maximum modularity in 10 random starts: 0.8588
Number of communities: 25
Elapsed time: 3 seconds
> DimPlot(luadobj, reduction="umap", group.by="seurat_clusters", pt.size=0.001, label=T)
```

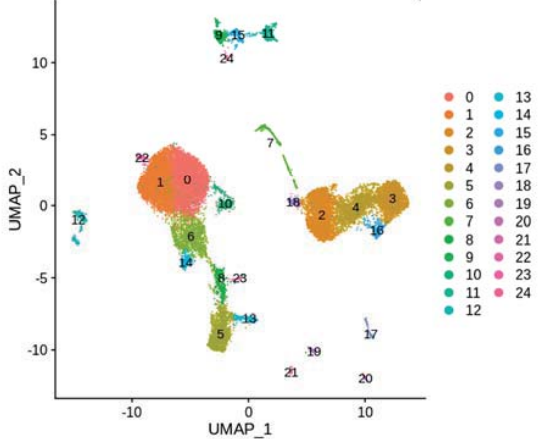






31

Identification of cluster-specific markers



32

Identification of cluster-specific markers

```
#MAST has good FDR control and is faster than DESeq2
luadobj.markers = FindAllMarkers(luadobj, only.pos=TRUE, min.pct=0.25, logfc.threshold=0.25, test.use="MAST") ;
```

`test.use` Denotes which test to use. Available options are:

- "wilcox" : Identifies differentially expressed genes between two groups of cells using a Wilcoxon Rank Sum test (default)
- "bimod" : Likelihood-ratio test for single cell gene expression, (McDavid et al., Bioinformatics, 2013)
- "roc" : Identifies 'markers' of gene expression using ROC analysis. For each gene, evaluates (using AUC) a classifier built on that gene alone, to classify between two groups of cells. An AUC value of 1 means that expression values for this gene alone can perfectly classify the two groupings (i.e. Each of the cells in cells.1 exhibit a higher level than each of the cells in cells.2). An AUC value of 0 also means there is perfect classification, but in the other direction. A value of 0.5 implies that the gene has no predictive power to classify the two groups. Returns a 'predictive power' (abs(AUC-0.5) * 2) ranked matrix of putative differentially expressed genes.
- "t" : Identify differentially expressed genes between two groups of cells using the Student's t-test.
- "negbinom" : Identifies differentially expressed genes between two groups of cells using a negative binomial generalized linear model. Use only for UMI-based datasets
- "poisson" : Identifies differentially expressed genes between two groups of cells using a poisson generalized linear model. Use only for UMI-based datasets
- "LR" : Uses a logistic regression framework to determine differentially expressed genes. Constructs a logistic regression model predicting group membership based on each feature individually and compares this to a null model with a likelihood ratio test.
- "MAST" : Identifies differentially expressed genes between two groups of cells using a hurdle model tailored to scRNA-seq data. Utilizes the MAST package to run the DE testing.
- "DESeq2" : Identifies differentially expressed genes between two groups of cells based on a model using DESeq2 which uses a negative binomial distribution (Love et al, Genome Biology, 2014).This test does not support pre-filtering of genes based on average difference (or percent detection rate) between cell groups. However, genes may be pre-filtered based on their minimum detection rate (min.pct) across both cell groups. To use this method, please install DESeq2, using the instructions at https://bioconductor.org/packages/release/bioc/html/DESeq2.html

33

Identification of cluster-specific markers

```
> head(luadobj.markers, 30)
               p_val avg_log2FC pct.1 pct.2 p_val_adj cluster       gene
INHBA              0   2.027373 0.957 0.324         0       0      INHBA
CCL20              0   1.948472 0.808 0.318         0       0      CCL20
CXCL3              0   1.835299 0.997 0.490         0       0      CXCL3
RND3               0   1.760417 0.711 0.210         0       0       RND3
TNF                0   1.717851 0.907 0.331         0       0        TNF
C1QA               0   1.593701 1.000 0.519         0       0       C1QA
IL1A               0   1.531175 0.692 0.180         0       0       IL1A
FBP1               0   1.530024 0.998 0.484         0       0       FBP1
FABP4              0   1.520869 0.967 0.406         0       0      FABP4
C1QB               0   1.489668 0.996 0.493         0       0       C1QB
CXCL5              0   1.463187 0.541 0.130         0       0      CXCL5
MCEMP1             0   1.349559 0.991 0.358         0       0     MCEMP1
SERPINA1           0   1.324133 0.998 0.501         0       0   SERPINA1
MRC1               0   1.294227 0.996 0.403         0       0       MRC1
ALDH2              0   1.290682 0.998 0.543         0       0      ALDH2
MARCO              0   1.281677 0.997 0.444         0       0      MARCO
SNX10              0   1.267388 0.989 0.432         0       0      SNX10
MS4A7              0   1.240686 0.998 0.449         0       0      MS4A7
VSIG4              0   1.233653 0.988 0.374         0       0      VSIG4
AC026369.3         0   1.210307 0.910 0.258         0       0 AC026369.3
LPL                0   1.186042 0.909 0.286         0       0        LPL
FTL                0   1.184815 1.000 0.996         0       0        FTL
C1QC               0   1.181961 0.989 0.374         0       0       C1QC
OLR1               0   1.164877 0.994 0.416         0       0       OLR1
STXBP2             0   1.144131 0.937 0.414         0       0     STXBP2
HLA-DRB5           0   1.140811 0.998 0.637         0       0   HLA-DRB5
LGALS3             0   1.119919 1.000 0.711         0       0     LGALS3
RETN               0   1.119641 0.794 0.301         0       0       RETN
MSR1               0   1.118809 0.983 0.391         0       0       MSR1
SERPING1           0   1.107077 0.944 0.352         0       0   SERPING1
```

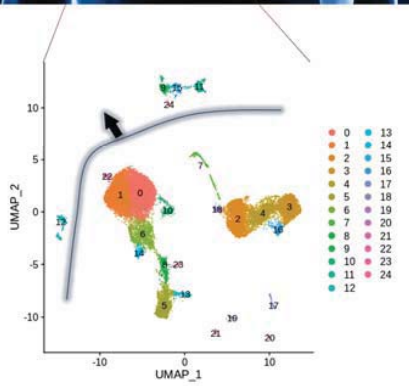
34

Identification of cluster-specific markers

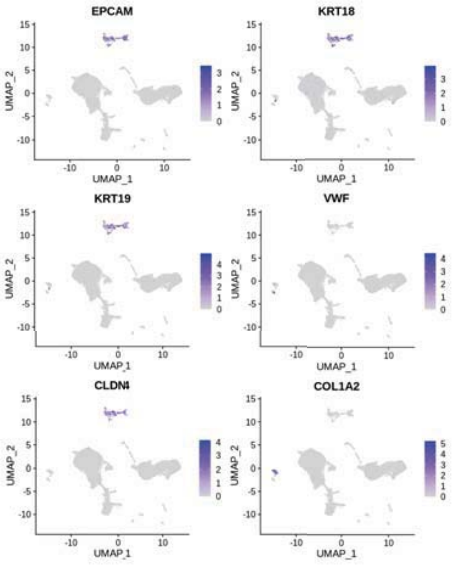
```
luadobj.markers.top20 = luadobj.markers %>% dplyr::group_by(cluster) %>% dplyr::top_n(n = 20, wt=avg_log2FC) ;

mySeuratClusters=unique(luadobj.markers.top20$cluster) ;

for(c in 1:length(mySeuratClusters)){
  luadobj.markers.top20.c = data.frame(
        cluster=luadobj.markers.top20[luadobj.markers.top20$cluster %in% mySeuratClusters[c], "gene"]) ;
  colnames(luadobj.markers.top20.c) = mySeuratClusters[c] ;
  if(c == 1){luadobj.markers.top20s = luadobj.markers.top20.c} else {
    luadobj.markers.top20s = cbind(luadobj.markers.top20s, luadobj.markers.top20.c)}
} ;
```

```
> luadobj.markers.top20s
          0       1       2       3       4       5       6       7       8       9      10      11      12      13      14      15      16      17      18      19      20      21      22      23      24
1     INHBA   FABP4     LTB   GNLY    GZMK    G0S2    CTSB   STMN1  FCER1A   SFTPC   NEAT1    CAPS     DCN    LST1    LGMN SCGB1A1    XCL1     IGKC    IL2RA    CPVL  TPSB2 PPP1R14B   HSPH1   DAPP1    EMP2
2     CCL20   APOC1     CD2    NKG7    CCL5    IL1B    EMP1    TUBB     CD1C  SFTPA2  GPCPD1 C20orf85     MGP  LILRB2  FOLR2 SCGB3A1    XCL2    CD79A  TNFRSF4   WDFY4   HPGDS   SEC61B    BAG3  CSF2RA ANKRD29
3     CXCL3   GCHFR   IL32  FGFBP2     CD8A  S100A9    LGMN   H2AFZ  CLEC10A  SFTPA1  ADAM17  C9orf24   FBLN1   COTL1   CCL13   WFDC2   KLRB1 TNFRSF13C TNFRSF18  C1orf54   VWA5A    TCF4  HSPA1A    FSCN1    AGER
4     RND3    MARCO     TRAC   MYOM2    GZMH    THBS1    CYBB   MKI67  HLA-DPB1   SFTPB    SNX2   TPPP3     CFD     LYN  RNASE1    CYB5A    JUND    BANK1     CTLA4   CLEC9A   TPSAB1    IRF7   HSPD1   GPR157    LMO7
5      TNF    C1QB     IL7R    GZMB    CD3D    CXCL8    CTSZ  TUBA1B   GPR183   NAPSA   ALOX5   RSPH1  CCDC80  NAMPT  MS4A6A   NCOA7   KLRC1     MS4A1     CD27     CST3    CPA3  CCDC50  DNAJB1    CCR7   RTKN2
6     C1QA  LGALS3     CD3D    PRF1    GZMA    TIMP1    TGFBI    TYMS   FCGR2B   SFTPD CCDC88A C11orf88  IGFBP7    SAT1    CTSB    PIGR     CD7     IGLC2     BATF     SNX3  MS4A2     PLD4    HSPA6    CCL22    CAV1
7      IL1A     GRN  ZFP36L2    KLRD1    NKG7  SERPINB2   LILRB4    PCLAF    RALA    SLPI  TNFAIP2  TSPAN1 SPARCL1    WARS SLC40A1    KRT7    CRTAM     IGHM     IL32    S100B   LTC4S    IRF4  HSPA1B   BIRC3 SLC39A8
8      FBP1    C1QA    CXCR4   SPON2   CXCR4     EREG C15orf48   HMGN2    CCL17     PGC SLC11A1  LRRIQ1  COL1A2   C5AR1 SELENOP   KRT19 TNFRSF18   JCHAIN    TIGIT HLA-DPB1     HDC    SOX4   HSPB1    LAMP3  GPRC5A
9     FABP4     FTL    TRBC2   CD247  TUBA4A    S100A8   MS4A6A   TOP2A C15orf48  SCGB3A2   MACC1     ELF3   TIMP3    AIF1    TGFBI     SLPI  FAM177A1  HERPUD1      TRAC  DNASE1L3   GATA2  LDLRAD4 HSP90AA1     TXN     AQP4
10     C1QB   ALDH2    DUSP4   PTGDS  PIK3R1     FCN1     ABL2      TK1  HLA-DPA1     MUC1     CAPG     AGR3     GSN    HES4    PLTP  TACSTD2    KLRD1     CD37 LINC01943    RGS10 SLC18A2    JCHAIN   DNAJA4    CCL19   CLIC5
11    CXCL5     ACPS     CD3G    CST7    IL32    PLAUR    TYMP   UBE2C    INSIG1     TCIM     TFRC     GSTA1   CALD1  CDKN1C    F13A1     ELF3     TRDC     IGHA1     CD3D     NAAA  TUBA1A    ITM2C     UBC     CD83    SCEL
12   MCEMP1    CCL18 LEPROTL1    CCL4    CD3G    PTGS2    FPR3   CENPF  SERPINB9  SLC34A2    GCHFR     PIFO    GNG11    FCN1    CTSZ SCGB3A2     CTSW     IGLC3      LTB     IRF8   IL1RL1  GPR183 HSP90AB1 MARCKSL1 TSPAN13
13 SERPINA1     SCD     CD3E    KLRF1    CST7    BASP1   MARCKS   NUSAP1    MS4A6A     CYB5A     GLUL    C5orf49    MFAP5  BCL2A1    IER3    CLDN4      REL     SSR4      CD2    LGALS2    RHEX   PMEPA1  ZFAND2A   MARCKS    CAV2
14     MRC1     CTSD   SPOCK2    KLRB1     TRAC     VCAN     SOD2    HMGB1    LGALS2     SFTA2     CSTB     CETN2  IGFBP6    TIMP1    SGK1    GPRC5A    PIK3R1   RALGPS2    PGM2L1    CPNE3   RGS13  HERPUD1    HMOX1    DUSP5     NTM
15    ALDH2     C1QC     RORA    CCL5    RUNX3     PPIF      FN1     PCNA  HLA-DQB1 C11orf96     IL1A    C2orf40     LUM    G0S2    EMP1     SOX4    IL2RB     IGHG3   ARID5B  HLA-DPA1     KIT C12orf75     IER5     ID2    VEGFA
16    MARCO    NUPR1    ARID5B    GZMH   TRGC2     SOD2     IER3      DEK     CD83     PIGR      FTL     SNTN     FBN1    FGL2     CCL2     LCN2     AREG     MZB1     ICOS  SERPINF1    CD69    GRASP   HSPE1    BASP1   CLDN18
17    SNX10     RBP4    CLEC2D    GZMA    DUSP2     AREG     APOE      DUT  HLA-DQA2   CXCL17    FABP5    PRDX5   COL1A1   PLAUR     CD14    SFTPB    IFITM2      LY9    TRBC2  TACSTD2     RGS2    NR4A3     UBB     CCL17   CYP4B1
18     MS4A7     CES1     CREM    CTSW   TRBC2  S100A12    GPNMB    PTTG1  HLA-DRA     ABCA3     RND3    MORN2    SFRP2    SOD2  CCL3L1   BPIFB1    SYTL3      EZR     DUSP4     FGL2     RGS1     RGS2      GRN   GPR183    KRT7
19    VSIG4 SERPING1  ANKRD12   CLIC3     CD3E     IER3   RNASE1    HMGB2     S100B     LAMP3     INHBA    DNAAF1  IGFBP5     IL1B     APOE   RNASE1    ZNF331      LTB    PMAIP1     SGK1      CLU     GZMB     CTSD  GADD45A CEACAM6
20     IFI27   CYP27A1     ETS1   PLAC8     MT2A    NAMPT    CCL18 HIST1H4C     G0S2     PEBP4      APOE  C1orf194    SFRP4 APOBEC3A    CCL18      MGP      GNLY     SMAP2     LAIR2     DUSP4     AREG     PTGDS MTRNR2L8 C15orf48    TIMP3
```

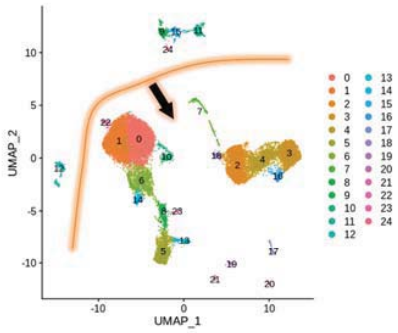

Discovery of cluster identity



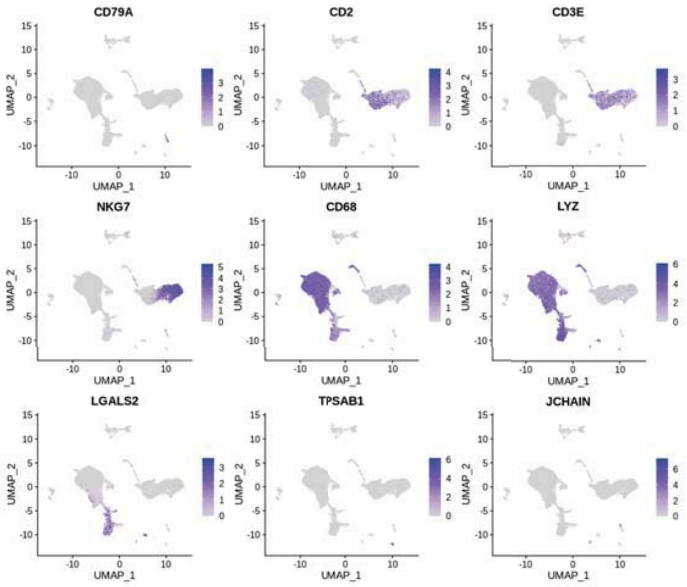

```
nonImm.markers = c("EPCAM","KRT18","KRT19","VWF","CLDN4","COL1A2") ;
FeaturePlot(luadobj, features=nonImm.markers, reduction="umap") ;
```



Discovery of cluster identity

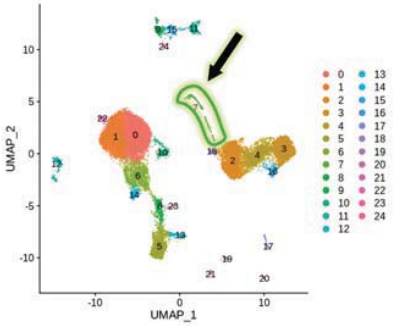



```
Imm.markers = c("CD79A","CD3E","NKG7","CD68","LYZ","LGALS2","TPSAB1","JCHAIN") ;
FeaturePlot(luadobj, features=Imm.markers, reduction="umap") ;
```

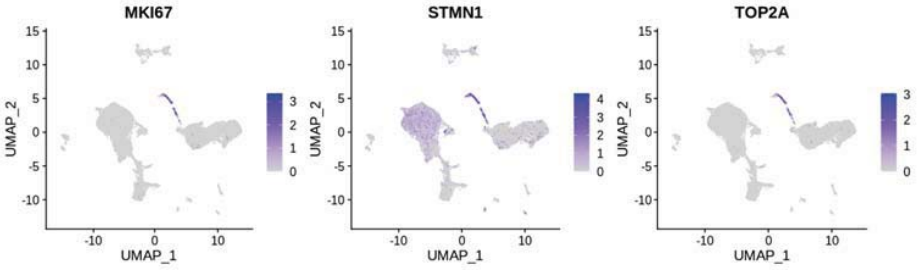


37

Discovery of cluster identity



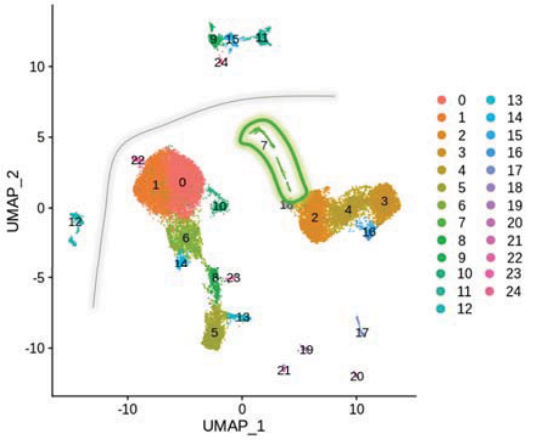

```
Proliferating.markers = c("MKI67","STMN1","TOP2A") ;
FeaturePlot(luadobj, features=Proliferating.markers, reduction="umap", ncol=3) ;
```

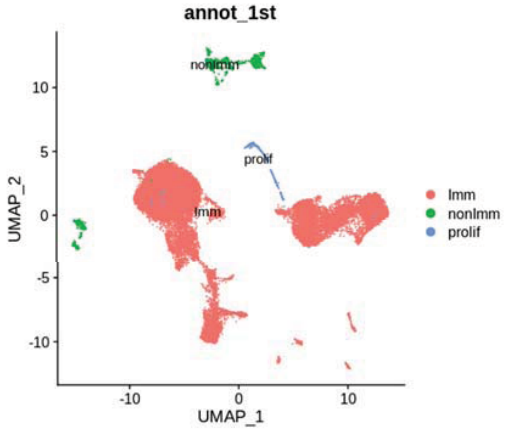


38

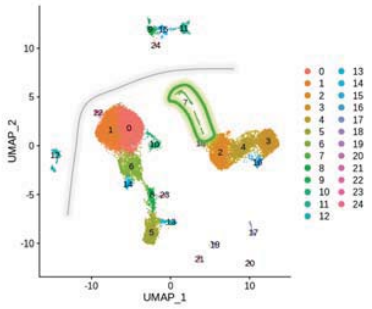
Discovery of cluster identity

```
luadobj$annot_1st = "" ;
luadobj@meta.data[luadobj@meta.data$seurat_clusters %in% c(9,11,15,24, 12), ]$annot_1st <- "nonImm" ;
luadobj@meta.data[luadobj@meta.data$seurat_clusters %in% c(0:6,8,10,13,14,16:23), ]$annot_1st <- "Imm" ;
luadobj@meta.data[luadobj@meta.data$seurat_clusters %in% c(7), ]$annot_1st <- "prolif" ;

Idents(luadobj) = "annot_1st" ;
DimPlot(luadobj, group.by = "annot_1st", reduction="umap", label=T) ;
```

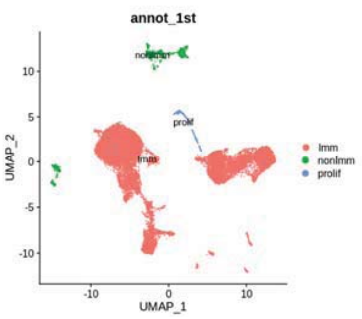


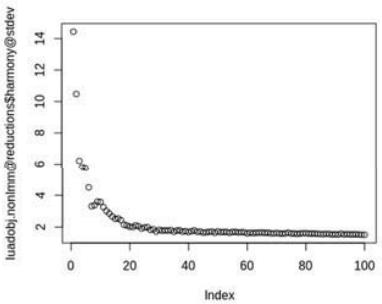


Discovery of sub-cluster identity



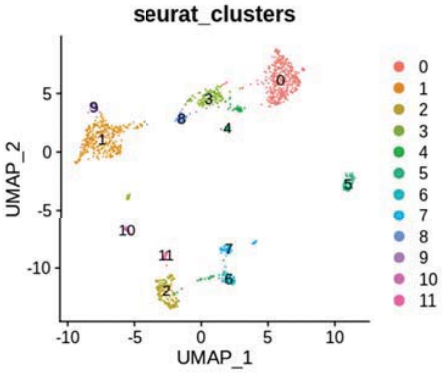

```
luadobj.nonImm = subset(luadobj, subset = seurat_clusters %in% c(9,11,15,24, 12)) ;

luadobj.nonImm = NormalizeData(luadobj.nonImm, normalization.method = "LogNormalize", scale.factor = 10000) ;
luadobj.nonImm = FindVariableFeatures(luadobj.nonImm, selection.method="vst", nfeatures=2000) ;
luadobj.nonImm = ScaleData(luadobj.nonImm, features=rownames(luadobj.nonImm)) ;
luadobj.nonImm = RunPCA(luadobj.nonImm, features=VariableFeatures(object=luadobj.nonImm), npcs=100) ;
luadobj.nonImm = RunHarmony(luadobj.nonImm, "orig.ident") ;
plot(luadobj.nonImm@reductions$harmony@stdev) ;
```



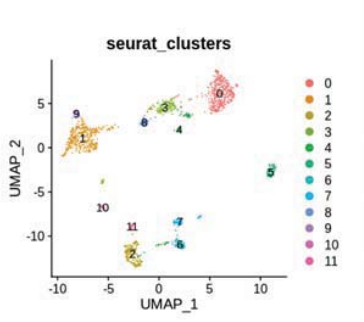


Discovery of sub-cluster identity

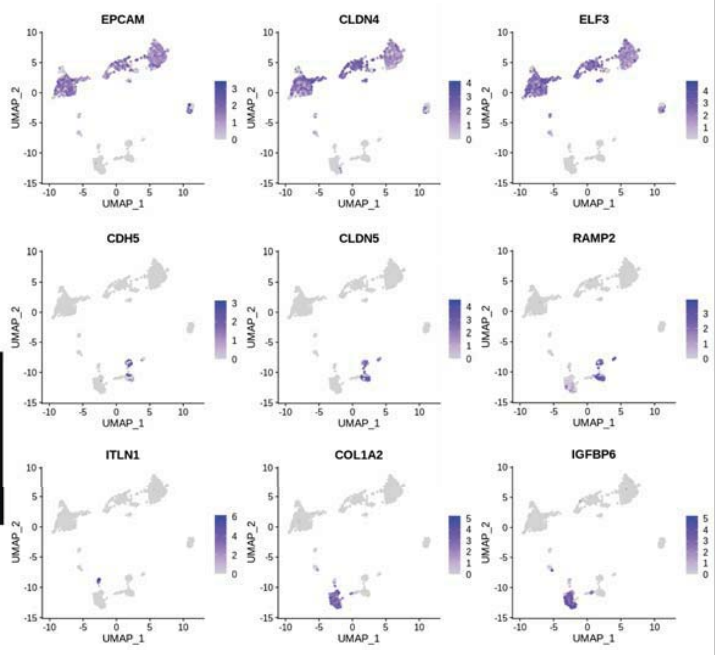
```
luadobj.nonImm = RunUMAP(luadobj.nonImm, reduction="harmony", dims=1:40, seed.use=1234) ;
luadobj.nonImm = RunTSNE(luadobj.nonImm, reduction="harmony", dims=1:40, seed.use=1234) ;
luadobj.nonImm = FindNeighbors(luadobj.nonImm, reduction="harmony", dims=1:40)
luadobj.nonImm = FindClusters(luadobj.nonImm, resolution=0.5) ;
DimPlot(luadobj.nonImm, reduction="umap", group.by="seurat_clusters", pt.size=0.001, label=T) ;
```



Discovery of sub-cluster identity

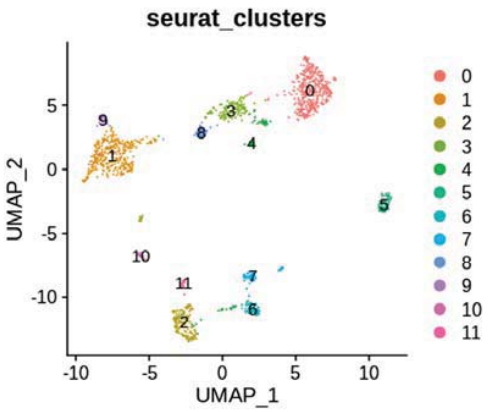


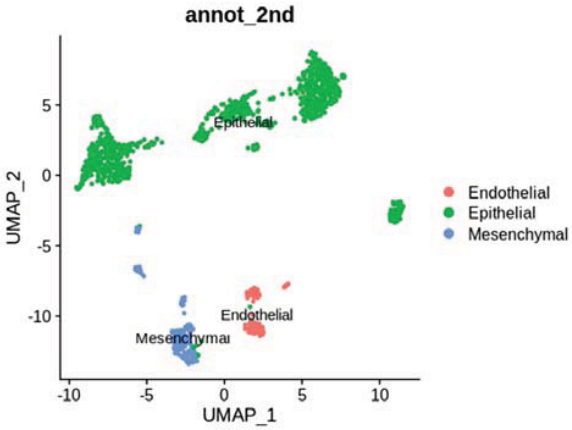

```
Epithelial.markers = c("EPCAM","CLDN4", "ELF3") ;
FeaturePlot(luadobj.nonImm, features=Epithelial.markers, reduction="umap", ncol=3) ;

Endothelial.markers = c("CDH5", "CLDN5", "RAMP2") ;
FeaturePlot(luadobj.nonImm, features=Endothelial.markers, reduction="umap", ncol=3) ;

Mesenchymal.markers = c("ITLN1","COL1A2","IGFBP6") ;
FeaturePlot(luadobj.nonImm, features=Mesenchymal.markers, reduction="umap", ncol=3) ;
```



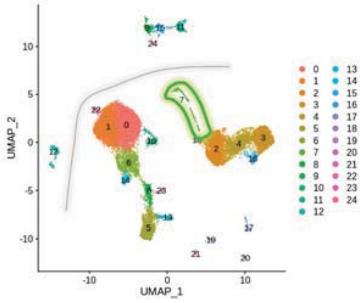
Discovery of sub-cluster identity

```
luadobj.nonImm$annot_2nd = "" ;
luadobj.nonImm@meta.data[luadobj.nonImm@meta.data$seurat_clusters %in% c(1,9,8,3,4,0,5), ]$annot_2nd <- "Epithelial" ;
luadobj.nonImm@meta.data[luadobj.nonImm@meta.data$seurat_clusters %in% c(6,7), ]$annot_2nd <- "Endothelial" ;
luadobj.nonImm@meta.data[luadobj.nonImm@meta.data$seurat_clusters %in% c(2,10,11), ]$annot_2nd <- "Mesenchymal" ;

Idents(luadobj.nonImm) = "annot_2nd" ;
DimPlot(luadobj.nonImm, group.by = "annot_2nd", reduction="umap", label=T) ;
```

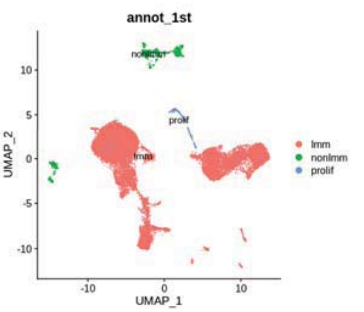


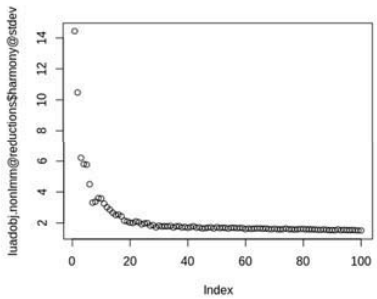


Discovery of sub-cluster identity



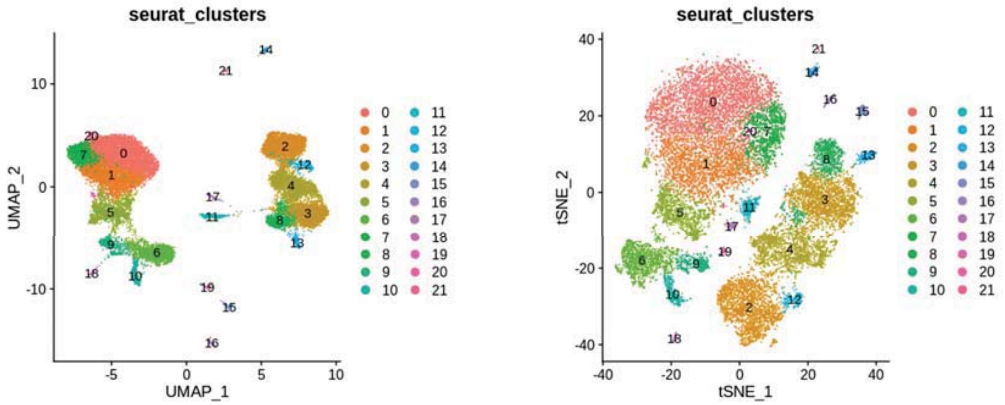

```
luadobj.Imm = subset(luadobj, subset = seurat_clusters %in% c(0:6,8,10,13,14,16:23)) ;

luadobj.Imm = NormalizeData(luadobj.Imm, normalization.method = "LogNormalize", scale.factor = 10000) ;
luadobj.Imm = FindVariableFeatures(luadobj.Imm, selection.method="vst", nfeatures=2000) ;
luadobj.Imm = ScaleData(luadobj.Imm, features=rownames(luadobj.Imm)) ;
luadobj.Imm = RunPCA(luadobj.Imm, features=VariableFeatures(object=luadobj.Imm), npcs=100) ;
luadobj.Imm = RunHarmony(luadobj.Imm, "orig.ident") ;
plot(luadobj.Imm@reductions$harmony@stdev) ;
```





Discovery of sub-cluster identity

```
luadobj.Imm = RunUMAP(luadobj.Imm, reduction="harmony", dims=1:50, seed.use=1234) ;
luadobj.Imm = RunTSNE(luadobj.Imm, reduction="harmony", dims=1:50, seed.use=1234) ;
luadobj.Imm = FindNeighbors(luadobj.Imm, reduction="harmony", dims=1:50)
luadobj.Imm = FindClusters(luadobj.Imm, resolution=0.8) ;
DimPlot(luadobj.Imm, reduction="umap", group.by="seurat_clusters", pt.size=0.001, label=T) ;
DimPlot(luadobj.Imm, reduction="tsne", group.by="seurat_clusters", pt.size=0.001, label=T) ;
```



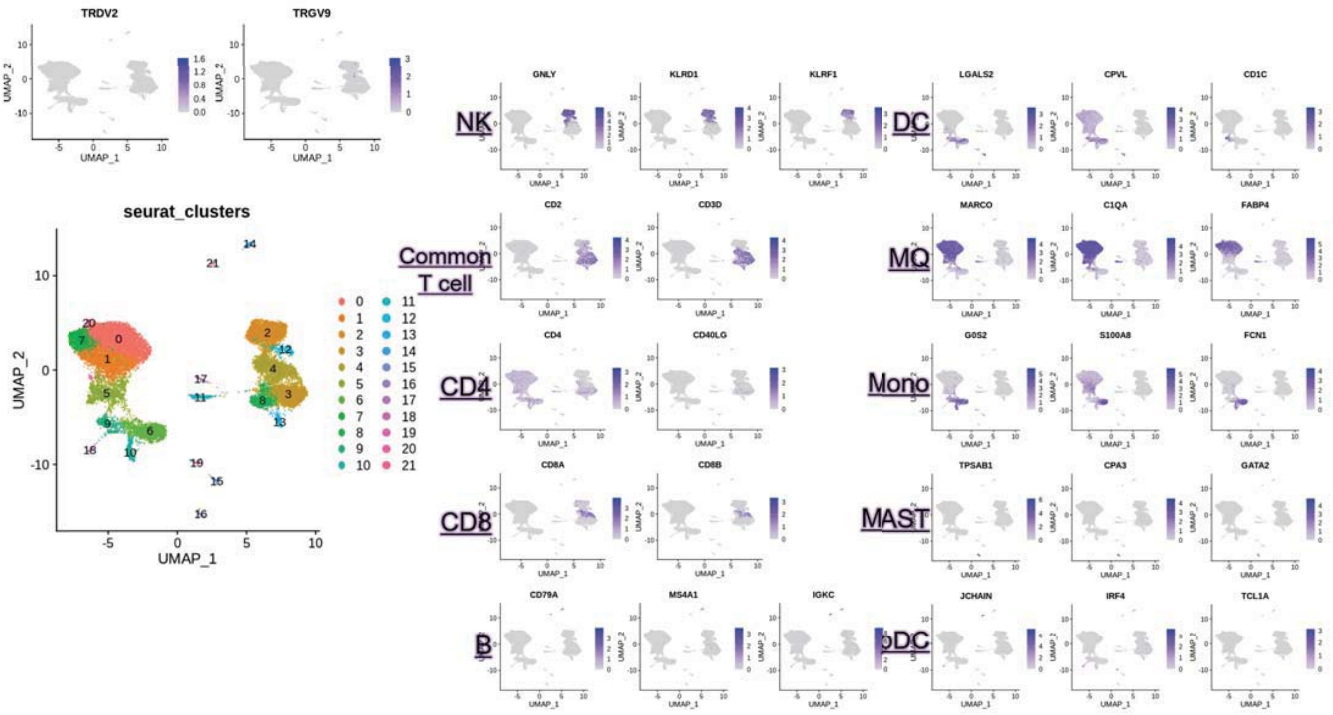
45

Discovery of sub-cluster identity

```
NK.markers = c("GNLY","KLRD1","KLRF1") ;
FeaturePlot(luadobj.Imm, features=NK.markers, reduction="umap", ncol=3) ;

T_common.markers = c("CD2","CD3D") ;
FeaturePlot(luadobj.Imm, features=T_common.markers, reduction="umap", ncol=3) ;

CD4.markers = c("CD4","CD40LG") ;
FeaturePlot(luadobj.Imm, features=CD4.markers, reduction="umap", ncol=3) ;

CD8.markers = c("CD8A","CD8B") ;
FeaturePlot(luadobj.Imm, features=CD8.markers, reduction="umap", ncol=3) ;

gdT.markers = c("TRDV2","TRGV9") ;
FeaturePlot(luadobj.Imm, features=gdT.markers, reduction="umap", ncol=3) ;

B.markers = c("CD79A","MS4A1","IGKC") ;
FeaturePlot(luadobj.Imm, features=B.markers, reduction="umap", ncol=3) ;

DC.markers = c("LGALS2","CPVL","CD1C") ;
FeaturePlot(luadobj.Imm, features=DC.markers, reduction="umap", ncol=3) ;

MQ.markers = c("MARCO","C1QA","FABP4") ;
FeaturePlot(luadobj.Imm, features=MQ.markers, reduction="umap", ncol=3) ;

Mono.markers = c("G0S2","S100A8","FCN1") ;
FeaturePlot(luadobj.Imm, features=Mono.markers, reduction="umap", ncol=3) ;

MAST.markers = c("TPSAB1","CPA3","GATA2") ;
FeaturePlot(luadobj.Imm, features=MAST.markers, reduction="umap", ncol=3) ;

pDC.markers = c("JCHAIN","IRF4","TCL1A") ;
FeaturePlot(luadobj.Imm, features=pDC.markers, reduction="umap", ncol=3) ;
```

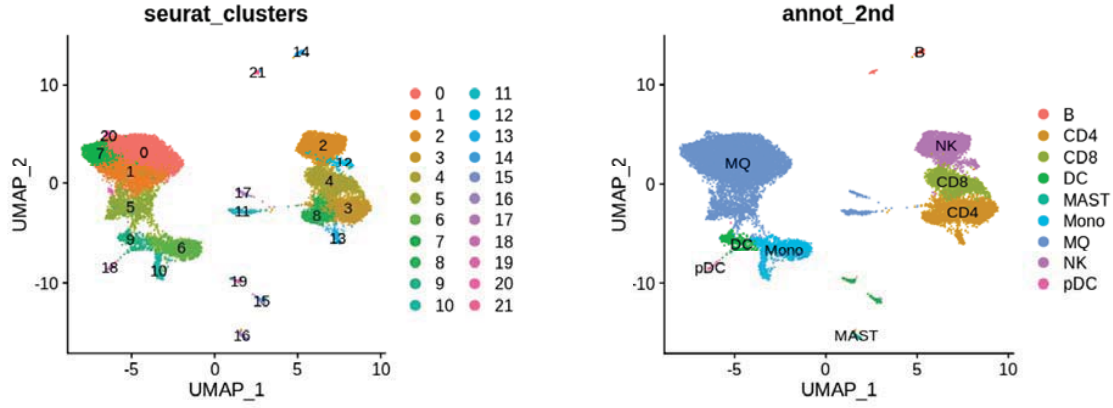
46

Discovery of sub-cluster identity



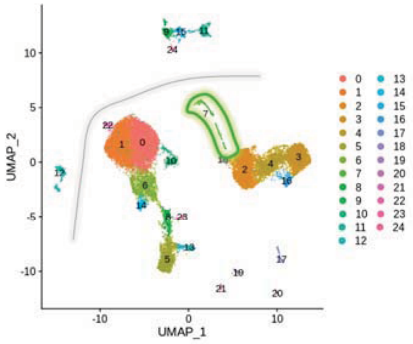
Discovery of sub-cluster identity

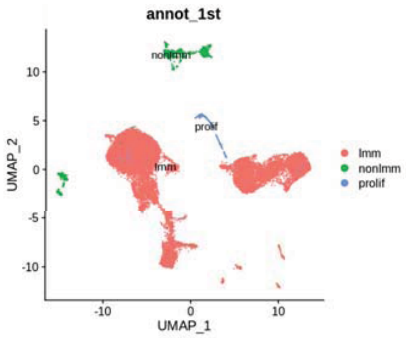
```
luadobj.Imm$annot_2nd = "" ;
luadobj.Imm@meta.data[luadobj.Imm@meta.data$seurat_clusters %in% c(2,12), ]$annot_2nd <- "NK" ;
luadobj.Imm@meta.data[luadobj.Imm@meta.data$seurat_clusters %in% c(3,8,13), ]$annot_2nd <- "CD4" ;
luadobj.Imm@meta.data[luadobj.Imm@meta.data$seurat_clusters %in% c(4), ]$annot_2nd <- "CD8" ;
luadobj.Imm@meta.data[luadobj.Imm@meta.data$seurat_clusters %in% c(14,21), ]$annot_2nd <- "B" ;
luadobj.Imm@meta.data[luadobj.Imm@meta.data$seurat_clusters %in% c(9,19,15), ]$annot_2nd <- "DC" ;
luadobj.Imm@meta.data[luadobj.Imm@meta.data$seurat_clusters %in% c(0,1,7,20,5,11,17), ]$annot_2nd <- "MQ" ;
luadobj.Imm@meta.data[luadobj.Imm@meta.data$seurat_clusters %in% c(6,10), ]$annot_2nd <- "Mono" ;
luadobj.Imm@meta.data[luadobj.Imm@meta.data$seurat_clusters %in% c(16), ]$annot_2nd <- "MAST" ;
luadobj.Imm@meta.data[luadobj.Imm@meta.data$seurat_clusters %in% c(18), ]$annot_2nd <- "pDC" ;

Idents(luadobj.Imm) = "annot_2nd" ;
DimPlot(luadobj.Imm, group.by = "annot_2nd", reduction="umap", label=T) ;
```

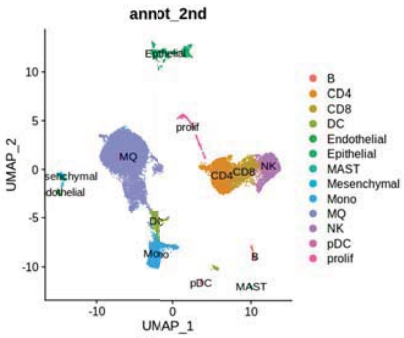


Discovery of sub-cluster identity

```
luadobj$annot_2nd = "" ;
luadobj@meta.data[luadobj@meta.data$annot_1st %in% "prolif",]$annot_2nd <- "prolif" ;
luadobj@meta.data[rownames(luadobj@meta.data) %in% rownames(luadobj.nonImm@meta.data),]$annot_2nd <- luadobj.nonImm@meta.data$annot_2nd ;
luadobj@meta.data[rownames(luadobj@meta.data) %in% rownames(luadobj.Imm@meta.data),]$annot_2nd <- luadobj.Imm@meta.data$annot_2nd ;
```







Thank you!

KIMQTAE@ajou.ac.kr